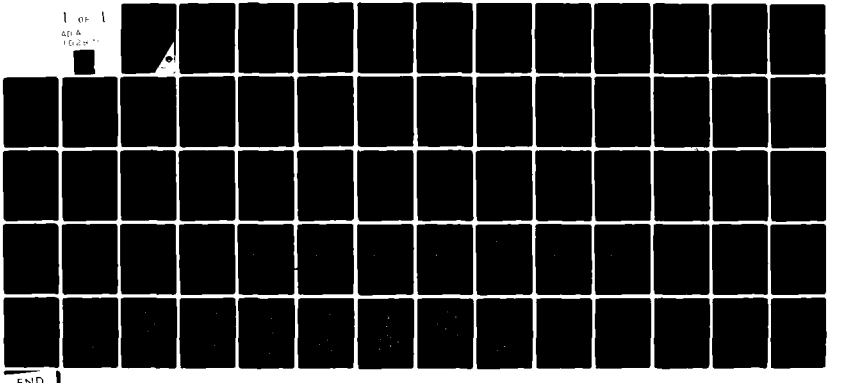
AD-A102 875 NAVAL OCEANOGRAPHIC OFFICE NSTL STATION MS F/G 22/2
A NEW TECHNIQUE FOR CORRECTING SATELLITE EPHEMERIS ERRORS INDIR--ETC(U)
APR 81 J R CLOUTIER
UNCLASSIFIED NOO-TR-246 NL



END
DATE FILMED
9-81
DTIC

TR 246

LEVEL II

TECHNICAL REPORT

A NEW TECHNIQUE FOR CORRECTING SATELLITE EPHEMERIS ERRORS INDIRECTLY OBSERVED FROM RADAR ALTIMETRY

JAMES R. CLOUTIER

APRIL 1981

PREPARED BY
COMMANDING OFFICER,
NAVAL OCEANOGRAPHIC OFFICE
NSTL STATION, BAY ST. LOUIS, MS 39522

PREPARED FOR
COMMANDER
NAVAL OCEANOGRAPHY COMMAND
NSTL STATION, BAY ST. LOUIS, MS 39529

FOREWORD

In the past three or four years, several methods (e.g., see References 1 and 2) have been devised and used to remove ephemeris errors present in geoid height and vertical deflection data obtained from the GEOS-3 satellite. The methods, though varying to some extent, all have two common features. These are that 1) the satellite's ground track is split into ascending and descending tracks and that 2) an *a priori* functional form is assumed for the ephemeris errors along each track.

There are several shortcomings in this approach. First, no *a priori* estimates of the ephemeris errors along each track are available. Lack of such information tends to make the selection of the functional form haphazard. Secondly, correlations between time-contiguous ascending and descending tracks are not accounted for. Failure to do so produces an unrealistic solution. Finally, discrete jumps in the ephemeris may be produced along solution-area boundaries. With regards to GEOS-3, this latter problem has never been addressed.

The technique described herein overcomes all of these drawbacks. It is believed that the use of this technique will resolve the inconsistencies observed in previously processed GEOS-3 data and enhance present processing of SEASAT data.

W. C. Palmer
Captain, USN
Commanding Officer

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NOO-TR-246	2. GOVT ACCESSION NO. AD-A102	3. RECIPIENT'S CATALOG NUMBER 875
4. TITLE (and Subtitle) A NEW TECHNIQUE FOR CORRECTING SATELLITE EPHEMERIS ERRORS INDIRECTLY OBSERVED FROM RADAR ALTIMETRY.		5. TYPE OF REPORT & PERIOD COVERED FINAL 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) James R. Cloutier		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Oceanographic Office NSTL Station Bay St. Louis, MS 39522		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Oceanographic Office NSTL Station Bay St. Louis, MS 39522		12. REPORT DATE APRIL 1981 13. NUMBER OF PAGES 64
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Radar altimetry, ground-track intersections, geoid height differences, ephemeris errors, conjugate gradient-projection algorithm, unbiased discrete function of minimum weighted variation, geoid height error, spline, non-unique solution, minimum curvature, and singular value decomposition.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

After high-frequency, temporal, and environmental effects have been removed from satellite radar-altimetry measurements, ephemeris (and other low-frequency) errors still remain. The vertical component of these low-frequency errors can be indirectly observed in the form of geoid height differences occurring at the satellite's ground-track intersections.

(Continued on reverse)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

250450

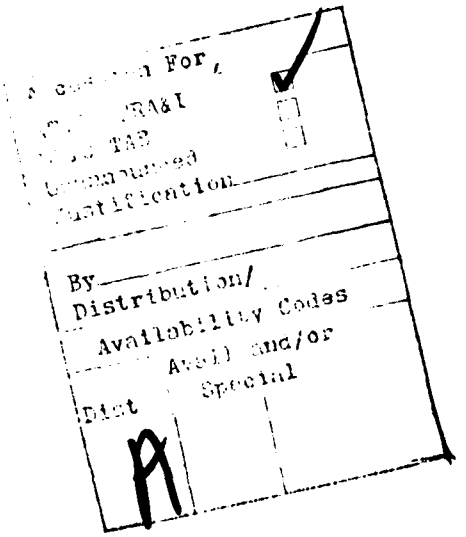
UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Abstract (Continued)

This report presents a new technique for correcting these errors. The main features of the technique are that 1) the satellite's ground-track is treated as one time-continuous track which repeatedly intersects itself and that 2) no *a priori* functional form of the ephemeris errors is assumed. A conjugate gradient-projection algorithm is used to find the unbiased, discrete function of minimum weighted variation which produces the geoid height differences observed at the ground-track intersections. After inspecting the discrete solution, a suitably-chosen continuous function can be fitted to the discrete data and used to reduce the ephemeris errors. Preliminary analysis of SEASAT geoid height data has indicated that an altitude correction to an accuracy of 20 centimeters rms is possible.

Although this new method is presently being applied to satellite data, the scope of the technique is much wider than this. The algorithm can be employed to reduce any time-dependent, low-frequency error present in network-type surveys (e.g., oceanographic shipboard and airborne surveys). Future applications include the removal of the diurnal variation present in magnetic surveys and the removal of the nonlinear gravimeter drift present in gravity surveys.



CONTENTS

LIST OF FIGURES

LIST OF FIGURES (Cont'd.)

LIST OF FIGURES (Cont'd.)

LIST OF TABLES

ACKNOWLEDGEMENTS

The author would like to thank Dr. Thomas M. Davis, Head Advanced Technology Office, for reviewing this report as well as Mr. William E. Rankin for his contributions during the project.

INTRODUCTION

The orbit of either the GEOS-3 or SEASAT satellite is such that, over a long period of time, the satellite's ground track repeatedly intersects itself. At many of these intersections, there is a data point in the form of an observed geoid height difference. Since high-frequency, temporal, and environmental effects have been previously removed from the satellite's radar-altimetry measurements, these observed differences are attributed to low-frequency errors in the satellite's ephemeris.[1] Let f(t) be the negative of the vertical component of the ephemeris error and let δ_{l_j} be the geoid height difference observed at the j^{th} intersection. We have[2]

$$f(t_{l_j}) - f(t_{p_j}) = \delta_{l_j}, \qquad j = 1,2,\ldots,N \tag{1}$$

where $t_{l_j} > t_{p_j}$ $\forall$ j and $t_{l_i} > t_{l_j}$ for i > j.

In order to correct the ephemeris errors, we must find a solution y(t) which satisfies Equation (1) either exactly or to some predetermined noise level.

Two facts complicate the situation. First, a bias in f(t) can never be detected through Equation (1) since it gets cancelled out in the subtraction process. Secondly, even within the equivalence class of all continuous functions with the same bias, the solution of Equation (1) is not unique. To overcome these difficulties, we seek an unbiased, low-frequency solution.

Keeping this in mind, there are two ways of approaching the problem. We can either assume a parameterized functional form $y(t, a_0, a_1, \ldots, a_n)$ for the solution and perform some type of optimization over the parameters or we can first seek a discrete solution $y_i = y(t_i)$ defined at the nodes t_{l_j}, t_{p_j}, j=1,..., N and then fit a continuous function to the result. Since no *a priori* information is available regarding the functional form, we choose the latter approach.

---

[1] Obviously, any long-wavelength errors will produce these differences. For simplicity, throughout this report we will classify all long-wavelength errors as ephemeris errors.

[2] The subscripts "l" and "p" are labels distinguishing later points in time from previous, or past, points in time.

STATEMENT OF THE PROBLEM

We can obtain an unbiased, discrete, low-frequency solution of Equation (1) by solving the following problem. Minimize the weighted[3] variation

$$J = \frac{1}{2} \sum_{i=1}^{2N-1} w_i (y_{i+1} - y_i)^2 \quad , \tag{2}$$

with respect to the discrete function y_i, $i = 1, \ldots, 2N$ subject to the constraints

$$y_{l_j} - y_{p_j} = \delta_{l_j} \qquad j = 1, \ldots, N \quad , \tag{3}$$

and the additional constraint

$$\sum_{i=1}^{2N} y_i = 0 \quad , \tag{4}$$

where[4]

$$y_i = y(t_i),\ y_{l_j} = y(t_{l_j}),\ y_{p_j} = y(t_{p_j}) \quad , \tag{5}$$

and

$$t_1 < t_2 < \ldots\ldots < t_{2N-1} < t_{2N} \tag{6}$$

Problem (2) - (4) can be simplified somewhat by solving Equation (3) for y_{l_j} and substituting the result into Equations (2) and (4). Before doing so, we define

$$k_i = \begin{cases} i \text{ if } \exists j \ni p_j = i \\ p_j \text{ if } \exists j \ni l_j = i \end{cases} \qquad i = 1, \ldots, 2N \quad , \tag{7}$$

and

$$\delta_{p_j} = 0 \qquad j = 1, \ldots, N \tag{8}$$

---

[3] The weights $w_i > 0$, $i = 1, \ldots, 2N-1$ will be used to account for the uneven spacing between intersections.

[4] It should be noted that for every i there exists a j such that either $l_j = i$ or $p_j = i$.

A new equivalent problem can now be stated.

Minimize the weighted variation

$$J = \frac{1}{2} \sum_{i=1}^{!!N-1} w_i (y_{k_{i+1}} - y_{k_i} + \delta_{i+1} - \delta_i)^2 \tag{9}$$

with respect to the discrete function y_{p_j}, $j = 1, \ldots,$ N subject to the constraint

$$\sum_{j=1}^{!!} y_{p_j} + \frac{1}{2} \sum_{j=1}^{!!} \delta_{1_j} = 0 \tag{10}$$

In vector notation this becomes minimize the quadratic function

$$f(\vec{z}) = \frac{1}{2} \vec{z}^T A \vec{z} + \vec{b}^T \vec{z} + c \quad , \tag{11}$$

with respect to $\vec{z}$, subject to the linear constraint

$$h(\vec{z}) = \vec{d}^T \vec{z} + e = 0 \quad , \tag{12}$$

where[5]

$$\vec{z} = \begin{bmatrix} y_{p_1} \\ y_{p_2} \\ \vdots \\ y_{p_N} \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{!!} \end{bmatrix}, \quad \vec{d} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix}, \qquad A = (a_{m,n}) \quad m = 1,\ldots,N \quad n = 1,\ldots,N \tag{13}$$

with

$$a_{m,m} = w_{p_m} + w_{p_{m-1}} + w_{1_m} + w_{1_{m-1}} \qquad m = 1,\ldots,N \tag{14}$$

---

[5] Since, from this point on, we will be dealing with the vector $\vec{z}$, it will be essential to remember that $z_j = y_{p_j}$, $j = 1, \ldots, N$.

$$a_{m,n} = -\sum_{i=1} w_i \quad \text{for } m \neq n \tag{15}$$

where

$$\hat{w}_1 = \begin{cases} w_{p_m} & \text{if } p_m = p_n - 1 \\ 0 & \text{otherwise} \end{cases} \tag{16-1}$$

$$\hat{w}_2 = \begin{cases} w_{p_n} & \text{if } p_m = p_n + 1 \\ 0 & \text{otherwise} \end{cases} \tag{16-2}$$

$$\hat{w}_3 = \begin{cases} w_{p_m} & \text{if } p_m = l_n - 1 \\ 0 & \text{otherwise} \end{cases} \tag{16-3}$$

$$\hat{w}_4 = \begin{cases} w_{l_n} & \text{if } p_m = l_n + 1 \\ 0 & \text{otherwise} \end{cases} \tag{16-4}$$

$$\hat{w}_5 = \begin{cases} w_{l_m} & \text{if } l_m = p_n - 1 \\ 0 & \text{otherwise} \end{cases} \tag{16-5}$$

$$\hat{w}_6 = \begin{cases} w_{p_n} & \text{if } l_m = p_n + 1 \\ 0 & \text{otherwise} \end{cases} \tag{16-6}$$

$$\hat{w}_7 = \begin{cases} w_{l_m} & \text{if } l_m = l_n - 1 \\ 0 & \text{otherwise} \end{cases} \tag{16-7}$$

$$\hat{w}_8 = \begin{cases} w_{l_n} & \text{if } l_m = l_n + 1 \\ 0 & \text{otherwise} \end{cases} \tag{16-8}$$

$$b_m = -w_{p_m - 1}\delta_{p_m - 1} - w_{p_m}\delta_{p_m + 1} + w_{l_m - 1}(\delta_{l_m} - \delta_{l_m - 1}) - w_{l_m}(\delta_{l_m + 1} - \delta_{l_m}) \quad m = 1, \ldots, N \tag{17}$$

$$c = \frac{1}{2} \sum_{i=1}^{2N-1} w_i(\delta_{i+1} - \delta_i)^2 \tag{18}$$

$$d_m = 1 \qquad m = 1,\ldots,N \tag{19}$$

$$e = \frac{1}{2} \sum_{j=1}^{N} \delta_{l_j} \tag{20}$$

In the above definitions,

$$\delta_0 = 0,\ \delta_{2N+1} = 0,\ w_0 = 0,\ w_{2N} = 0 \tag{21-1}$$

and, in Equations (14) - (17),

$$w_{p_m} = 0 \qquad \text{if } l_m = p_m + 1 \tag{21-2}$$

We now turn our attention to the solution of Problem (11) - (12).

COMMENTS ON THE SOLUTION OF THE PROBLEM

We are confronted with the problem of minimizing the quadratic function[6]

$$f(\vec{z}) = \frac{1}{2}\,\vec{z}^{T}A\vec{z} + \vec{b}^{T}\vec{z} \tag{22}$$

subject to the linear constraint

$$h(\vec{z}) = \vec{d}^{T}\vec{z} + e = 0 \tag{23}$$

This problem has a unique global solution providing that the matrix A is positive definite on the subspace $M_1 = \{\vec{z}:\vec{d}^{T}\vec{z} = 0\}$.[7] The problem can be recast as that of minimizing the Lagrangian

$$F(\vec{z},\lambda) = f(\vec{z}) + \lambda h(\vec{z}) \tag{24-1}$$

$$= \frac{1}{2}\,\vec{z}^{T}A\vec{z} + \vec{b}^{T}\vec{z} + \lambda(\vec{d}^{T}\vec{z} + e) \tag{24-2}$$

with respect to the vector $\vec{z}$ subject to the constraint (23), and the solution can be found by solving the linear system

$$F_{\vec{z}}(\vec{z},\lambda) = \vec{0} \tag{25-1}$$

$$h(\vec{z}) = 0 \tag{25-2}$$

Specifically, we must find the vector $\vec{z}$ and the scalar Lagrange multiplier λ which satisfy

$$\left[\begin{array}{c|c} A & \vec{d} \\ \hline \vec{d}^{T} & 0 \end{array}\right]\left[\begin{array}{c} \vec{z} \\ \hline \lambda \end{array}\right] = \left[\begin{array}{c} -\vec{b} \\ \hline -e \end{array}\right] \tag{26}$$

This seems simple and straightforward, and is indeed, providing N, the number of track intersections, is relatively small. Large segments of the track, however, may contain thousands of intersections, making the direct solution of Equation (26) extremely

---

[6] Here, the third term c has been dropped from $f(\vec{z})$ since it has no effect on the solution.

[7] The positive definiteness of A is discussed in Appendix A.

difficult if not impossible. In order to have the capability to process such large segments, we shun any direct method of solving (26) and select an iterative method, the conjugate gradient-projection algorithm. In doing so, we will be able to take full advantage of the fact that the matrix A is sparse (see Appendix A).

THE CONJUGATE GRADIENT-PROJECTION ALGORITHM

Conjugate gradient algorithms are well documented in the literature. References 3 and 4 provide particularly lucid treatises of the subject. These algorithms differ from ordinary gradient methods in that at each step, instead of moving in the negative direction of the gradient, movement is made along a direction $\vec{p}_k$ which is Q-orthogonal to all previous step directions $\vec{p}_l$, i.e.,

$$\vec{p}_k^T Q \vec{p}_l = 0, \qquad l = 0,1,\ldots,k-1 \tag{27}$$

where Q is a symmetric, positive definite matrix.[8]

For unconstrained problems, the direction $\vec{p}_k$ is obtained from a linear combination of the present gradient $\vec{g}_k$ and the previous search direction $\vec{p}_{k-1}$,

$$\vec{p}_k = -\vec{g}_k + \beta_k \vec{p}_{k-1} \tag{28}$$

The directional coefficient β_k is chosen to satisfy the conjugacy requirement (27).

For linearly constrained problems, the direction $\vec{p}_k$ is obtained as above with the exception that the gradient is replaced by its projection onto the subspace defined by the constraints—hence, the nomenclature gradient-projection. This procedure enables satisfaction of the constraints at every iteration.

To introduce our particular algorithm, let $\vec{z}$ denote a nominal point satisfying Equation (23), let $\vec{z}'$ denote a varied point, and let $\vec{p}$ denote a search direction which, when multiplied by the stepsize α, leads from the nominal point to the varied point. Then, by definition,

$$\vec{z}' = \vec{z} + \alpha\vec{p} \tag{29}$$

Furthermore, let the present search direction $\vec{p}$ be related to the previous search direction $\hat{\vec{p}}$ via

$$\vec{p} = -F_{\vec{z}}(\vec{z},\lambda) + \beta\hat{\vec{p}} \tag{30}$$

Equations (29) and (30) contain three unknown parameters - the Lagrange multiplier λ, the directional coefficient β, and the stepsize α. We will determine these parameters so that 1) the constraint (23) is satisfied at every iteration, 2) the present search direction is A-orthogonal to the previous search direction and 3) the maximum amount of decrease in the function $f(\vec{z})$ is attained at each iteration.

---

[8] Vectors $\vec{p}_i$, i = 0, 1, ..., k satisfying Equation (27) are said to be conjugate with respect to Q.

For the constraint (23) to be satisfied, we must have

$$h(\vec{z}') = 0 \tag{31}$$

Since we have assumed $\vec{z}$ satisfies (23), this implies that

$$\frac{1}{\alpha}\ \Delta h(\vec{z}\ ) = \frac{1}{\alpha}\left[h(\vec{z}') - h(\vec{z})\right] = 0 \tag{32-1}$$

$$= \frac{1}{\alpha}\left[(\vec{z}'^{T}\vec{d} + e) - (\vec{z}^{T}\vec{d} + e)\right] = 0 \tag{32-2}$$

$$= \frac{1}{\alpha}\ (\vec{z}' - \vec{z})^{T}\vec{d} = 0 \tag{32-3}$$

$$= \vec{p}^{T}\vec{d} = 0 \tag{32-4}$$

Substituting Equation (30) into Equation (32-4) produces

$$(-F_{\vec{z}}(\vec{z},\lambda) + \beta\hat{p})^{T}\vec{d} = 0 \tag{33}$$

Providing that we have forced $\hat{p}^{T}\vec{d}$ to be equal to zero in the previous iteration, this reduces to

$$F_{\vec{z}}^{T}(\vec{z},\lambda)\vec{d} = 0 \tag{34-1}$$

or

$$(A\vec{z} + \vec{b} + \lambda\vec{d})^{T}\vec{d} = 0 \tag{34-2}$$

We now choose λ so that Equation (34-2) is satisfied. We have

$$\lambda = -(A\vec{z} + \vec{b})^{T}\vec{d}/\vec{d}^{T}\vec{d} \tag{35-1}$$

$$= -f_{\vec{z}}^{T}(\vec{z})\vec{d}/\vec{d}^{T}\vec{d} \tag{35-2}$$

$$= -\frac{1}{N}\sum_{k=1}^{N} f_{z_k}(\vec{z}) \tag{35-3}$$

To prevent the above logic from breaking down, on the first iteration we set $\beta = 0$.

For the present search direction $\vec{p}$ to be A-orthogonal to the previous search direction $\hat{p}$, we must have

$$\vec{p}^T A\hat{p} = 0 \tag{36}$$

Coupled with Equation (30), this implies that

$$-F_{\vec{z}}^T(\vec{z},\lambda)A\hat{p} + \beta\hat{p}^T A\hat{p} = 0 \tag{37-1}$$

or

$$\beta = \frac{F_{\vec{z}}^T(\vec{z},\lambda)A\hat{p}}{\hat{p}^T A\hat{p}} \tag{37-2}$$

To achieve the maximum amount of decrease in the function $f(\vec{z})$ at each iteration we first note that with λ given by (35-3),

$$f(\vec{z}') \equiv F(\vec{z}',\lambda) \tag{38}$$

Furthermore, since both λ and β have been specified, $F(\vec{z}',\lambda)$ is a function of α only.

$$F(\alpha) = F(\vec{z}',\lambda) \tag{39-1}$$

$$= F(\vec{z} + \alpha\vec{p},\lambda) \tag{39-2}$$

$$= \frac{1}{2}(\vec{z} + \alpha\vec{p})^T A(\vec{z} + \alpha\vec{p}) + \vec{b}^T(\vec{z} + \alpha\vec{p}) + \lambda[\vec{d}^T(\vec{z} + \alpha\vec{p}) + e] \tag{39-3}$$

$$= (\frac{1}{2}\vec{p}^T A\vec{p})\alpha^2 + [(A\vec{z} + \vec{b} + \lambda\vec{d})^T\vec{p}]\alpha + \frac{1}{2}\vec{z}^T A\vec{z} + \vec{b}^T\vec{z} + \lambda(\vec{d}^T\vec{z} + e) \tag{39-4}$$

$$= (\frac{1}{2}\vec{p}^T A\vec{p})\alpha^2 + F_{\vec{z}}^T(\vec{z},\lambda)\vec{p}\alpha + F(0) \tag{39-5}$$

Minimizing Equiation (39-5) with respect to α, we have

$$F_\alpha(\alpha) = \vec{p}^T A\vec{p}\alpha + F_{\vec{z}}^T(\vec{z},\lambda)\vec{p} = 0 \tag{40}$$

so that the optimum stepsize is given by

$$\alpha = -F_{\vec{z}}^{T}(\vec{z},\cdot)\vec{p}/\vec{p}^{T}A\vec{p} \tag{41}$$

Note that for a minimum we must have

$$F_{\alpha\alpha}(\alpha) = \vec{p}^{T}A\vec{p} > 0 \tag{42}$$

which will always be satisfied if A is positive definite on the subspace M_1. (Refer Equation (32-4).)

The bare basics of the algorithm have now been derived. Many additional properties of the algorithm exist. Perhaps the most important is that, when applied to a quadratic-linear problem such as (22) - (23), convergence is assured in at most N-q iterations, where q is the dimension of the constraint h. Other properties, such as additional orthogonality properties, will not be discussed here. We will state, however, that due to these additional orthogonality properties, Equations (37-2) and (41) can be reduced to

$$\beta = Q(\vec{z},\cdot)/Q(\hat{z},\hat{\cdot}) \tag{43}$$

and

$$\alpha = Q(\vec{z},\lambda)/\vec{p}^{T}A\vec{p} \tag{44}$$

where

$$Q(\vec{z},\lambda) = F_{\vec{z}}^{T}(\vec{z},\lambda)F_{\vec{z}}(\vec{z},\lambda) \tag{45}$$

The symbol "^" refers to quantities computed in the previous iteration.

SUMMARY OF THE ALGORITHM

The conjugate gradient-projection algorithm has now been fully specified. The matrix A and the vector $\vec{b}$ need to be computed only once.[9] After this is done, the algorithm is:

Step 1. Select a nominal point $\vec{z}$ such that

$$h(\vec{z}) = \vec{d}^T\vec{z} + e = 0 \tag{46}$$

Step 2. Compute

$$f_{\vec{z}}(\vec{z}) = A\vec{z} + \vec{b} \tag{47}$$

Step 3. Compute

$$= -\frac{1}{N}\sum_{k=}^{N} f_{z_k}(\vec{z}) \tag{48}$$

Step 4. Compute

$$F_{\vec{z}}(\vec{z},\cdot) = f_{\vec{z}}(\vec{z}) + \lambda\vec{d} \tag{49}$$

Step 5. Compute

$$Q(\vec{z},\cdot) = F_{\vec{z}}^T(\vec{z},\lambda)F_{\vec{z}}(\vec{z},\lambda) \tag{50}$$

Step 6. Check for convergence. Is

$$\sqrt{Q(\vec{z},\lambda)/\vec{z}^T\vec{z}} \leq \epsilon \tag{51}$$

where ϵ is some small, preselected number. If (51) is satisfied, stop; if not, proceed with step 7.

Step 7. Is the iteration number equal to N-q? If so, stop; if not, proceed with step 8.

Step 8. Compute the directional coefficient β. If this is the first iteration, $\beta = 0$; otherwise

---

[9] The matrix A and the vector $\vec{b}$ require 6 x N storage locations.

$$\beta = Q(\vec{z},\lambda)/Q(\hat{z},\hat{\lambda}) \tag{52}$$

Step 9. Determine the search direction

$$\vec{p} = -F_{\vec{z}}(\vec{z},\lambda) + \beta\hat{p} \tag{53}$$

Step 10. Compute the optimum stepsize

$$\alpha = Q(\vec{z},\lambda)/\vec{p}^{T}A\vec{p} \tag{54}$$

Step 11. Compute the varied point

$$\vec{z}' = \vec{z} + \alpha\vec{p} \tag{55}$$

Step 12. Replace the old nominal values $\vec{z}$, $\hat{p}$, and $Q(\hat{z},\hat{\lambda})$ with the new nominal values $\vec{z}'$, $\vec{p}$, and $Q(\vec{z},\lambda)$

$$\vec{z} = \vec{z}' \tag{56-1}$$

$$\hat{p} = \vec{p} \tag{56-2}$$

$$Q(\hat{z},\hat{\lambda}) = Q(\vec{z},\lambda) \tag{56-3}$$

and proceed to step 2.

Once convergence has been attained, the remaining half of the ephemeris errors are computed via Equation (3).

ON CORRECTING GEOID HEIGHT ERRORS RELATIVE TO A REFERENCE SURFACE

This section deals with the situation in which segments of the satellite track have been previously corrected and it is desired to correct the remaining segments relative to these. In such a situation, any of the nodes y_{l_j} or y_{p_j} falling in a previously corrected segment are deemed to be errorless. Let I_o be the index set of the errorless nodes. Then

$$y_{l_j} = 0 \text{ for } l_j \in I_o \tag{57-1}$$

$$y_{p_j} = 0 \text{ for } p_j \in I_o \tag{57-2}$$

Since

$$y_{l_j} - y_{p_j} = \delta_{l_j} \quad j = 1,\ldots,N \tag{58}$$

we must have

$$y_{p_j} = -\delta_{l_j} \text{ for } l_j \in I_o, p_j \notin I_o \tag{59}$$

Let us now see how these constraints, namely Equations (57-2) and (59), affect our original problem. Refer to Equations (9) and (10). We want to minimize the weighted variation J subject not only to the constraint (10), but also to the additional constraints (57-2) and (59).

At this point, the straightforward approach would be to substitute Equations (57-2) and (59) into Equations (9) and (10). This would create a simpler, equivalent problem of lower order. Unfortunately, it would also create a complicated indexing problem.

Without this substitution, Problem (11) - (12) becomes minimize the quadratic function

$$f(\vec{z}) = \frac{1}{2}\vec{z}^T A\vec{z} + \vec{b}^T\vec{z} + c \tag{60}$$

with respect to $\vec{z}$, subject to the linear constraint

$$\vec{h}(\vec{z}) = C^T\vec{z} + \vec{g} = \vec{0} \tag{61}$$

where $\vec{z}$, $\vec{b}$, c, and A are defined as before and where the N x (m+1)-matrix C and the (m+1)-vector $\vec{g}$ are given by

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & \cdot & \cdot & & & \cdot \\ 1 & \cdot & \cdot & & & \cdot \\ 1 & 0 & \cdot & & & \cdot \\ 1 & 1 & \cdot & & & \cdot \\ \cdot & 0 & \cdot & & & \cdot \\ \cdot & \cdot & \cdot & & & \cdot \\ \cdot & \cdot & \cdot & & & 0 \\ \cdot & \cdot & \cdot & & & 1 \\ \cdot & \cdot & 0 & & & 0 \\ \cdot & \cdot & 1 & & & \cdot \\ \cdot & \cdot & 0 & & & \cdot \\ & & \cdot & & & \cdot \\ & & \cdot & & & \cdot \\ 1 & 0 & 0 & \cdots & \cdots & 0 \end{bmatrix}, \quad \vec{g} = \begin{bmatrix} \frac{1}{2} \sum_{j=1}^{N} l_j \\ g_{j_1} \\ g_{j_2} \\ g_{j_3} \\ \cdot \\ \cdot \\ \cdot \\ g_{j_m} \end{bmatrix} \tag{62}$$

with

$$g_{j_i} = \begin{cases} l_{j_i} & \text{if } l_{j_i} < I_c \text{ and } p_{j_i} > I_c \\ 0 & \text{if } p_{j_i} < I_c \end{cases} \tag{63}$$

Here, m is the number of errorless nodes. The first column of the C matrix contains all ones; the remaining columns each contain only one nonzero element whose value is one.

Problem (60) - (61) has a unique global solution providing that the matrix A is positive definite on the subspace $M_2 = \{\vec{z}: C^T\vec{z} = \vec{0}\}$. The problem can be recast as that of minimizing the Langrangian[10]

$$F(\vec{z}, \vec{\lambda}) = f(\vec{z}) + \vec{\lambda}^T\vec{h}(\vec{z}) \tag{64-1}$$

$$= \frac{1}{2}\vec{z}^TA\vec{z} + \vec{b}^T\vec{z} + \vec{\lambda}^T(C^T\vec{z} + \vec{g}) \tag{64-2}$$

with respect to the vector $\vec{z}$ subject to the constraint (61), and the solution can be found by solving the linear system

$$F_{\vec{z}}(\vec{z}, \vec{\lambda}) = \vec{0} \tag{65-1}$$

$$\vec{h}(\vec{z}) = \vec{0} \tag{65-2}$$

---

[10] Again, the third term c has been dropped from $f(\vec{z})$ since it has no effect on the solution.

for the vector $\vec{z}$ and the vector Lagrange multiplier $\vec{\lambda}$.

Referring back to the two previous sections, we can trace the modifications to the conjugate gradient-projection algorithm which will be required to solve the system (65). We have:

$$\vec{p} = -F_{\vec{z}}(\vec{z},\vec{\lambda}) + \beta\hat{p} \tag{66}$$

instead of Equation (30);

$$C^T\vec{p} = \vec{0} \tag{67}$$

instead of Equation (32-4);

$$C^T(-F_{\vec{z}}(\vec{z},\vec{\lambda}) + \beta\hat{p}) = \vec{0} \tag{68}$$

instead of Equation (33);

$$C^T(A\vec{z} + \vec{b} + C\vec{\lambda}) = \vec{0} \tag{69}$$

instead of Equation (34-2);

$$\vec{\lambda} = -(C^TC)^{-1}C^T(A\vec{z} + \vec{b}) \tag{70}$$

instead of Equation (35-1);

$$F_{\vec{z}}(\vec{z},\vec{\lambda}) = f_{\vec{z}}(\vec{z}) + C\vec{\lambda} \tag{71}$$

instead of Equation (49).

Since C^TC is an $(m+1)x(m+1)$ -matrix, the main concern now is to try to simplify Equation (70) by solving for the inverse of

$$C^TC = \left[\begin{array}{c|ccccc} N & 1 & 1 & 1 & \cdots & 1 \\ \hline 1 & 1 & 0 & \cdots & \cdots & 0 \\ 1 & 0 & \ddots & & & \vdots \\ 1 & \vdots & \ddots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \vdots \\ \vdots & \vdots & & & \ddots & 0 \\ 1 & 0 & \cdots & \cdots & 0 & 1 \end{array}\right] = \left[\begin{array}{c|c} N & \vec{u}^T \\ \hline \vec{u} & I \end{array}\right] \tag{72}$$

analytically. We need to find γ , $\vec{v}$, and W such that

$$\left[\begin{array}{c|c} N & \vec{u}^T \\ \hline \vec{u} & I \end{array}\right] \left[\begin{array}{c|c} \gamma & \vec{v}^T \\ \hline \vec{v} & W \end{array}\right] = \left[\begin{array}{c|c} 1 & \vec{0}^T \\ \hline \vec{0} & I \end{array}\right] \tag{73}$$

Expanding we have

$$\gamma N + \vec{u}^T\vec{v} = 1 \tag{74-1}$$

$$N\vec{v}^T + \vec{u}^T W = \vec{0}^T \tag{74-2}$$

$$\gamma\vec{u} + \vec{v} = \vec{0} \tag{74-3}$$

$$\vec{u}\vec{v}^T + W = I \tag{74-4}$$

Equations (74-1) and (74-3) imply that

$$\gamma = \frac{1}{N - m} \tag{75}$$

$$\vec{v} = -\frac{1}{N - m}\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{76}$$

while Equation (74-4) implies that

$$W = I - \vec{u}\vec{v}^T = \frac{1}{N - m}\begin{bmatrix} N - m + 1 & 1 & 1 & \cdots & 1 \\ 1 & \ddots & & & \vdots \\ 1 & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ 1 & \cdots & 1 & N - m + 1 & 1 \end{bmatrix} \tag{77}$$

Thus

$$(C^T C)^{-1} = \frac{1}{N-m}\begin{bmatrix} 1 & -1 & -1 & \cdots & -1 \\ -1 & N-m+1 & 1 & \cdots & 1 \\ -1 & 1 & N-m+1 & \ddots & 1 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -1 & 1 & \cdots & 1 & N-m+1 \end{bmatrix} \tag{78}$$

Returning to Equation (70) we have that

$$C^T(A\vec{z} + \vec{b}) = \begin{bmatrix} \sum_{j=1}^{N} f_{z_j}(\vec{z}) \\ f_{z_{j_1}}(\vec{z}) \\ f_{z_{j_2}}(\vec{z}) \\ \vdots \\ f_{z_{j_m}}(\vec{z}) \end{bmatrix} \tag{79}$$

so that

$$\vec{\lambda} = -\frac{1}{N-m}\begin{bmatrix} \sum_{j=1}^{N} f_{z_j}(\vec{z}) - \sum_{i=1}^{m} f_{z_{j_i}}(\vec{z}) \\ -\left(\sum_{j=1}^{N} f_{z_j}(\vec{z}) - \sum_{i=1}^{m} f_{z_{j_i}}(\vec{z})\right) + (N-m) f_{z_{j_1}}(\vec{z}) \\ -\left(\sum_{j=1}^{N} f_{z_j}(\vec{z}) - \sum_{i=1}^{m} f_{z_{j_i}}(\vec{z})\right) + (N-m) f_{z_{j_2}}(\vec{z}) \\ \vdots \\ -\left(\sum_{j=1}^{N} f_{z_j}(\vec{z}) - \sum_{i=1}^{m} f_{z_{j_i}}(\vec{z})\right) + (N-m) f_{z_{j_m}}(\vec{z}) \end{bmatrix} \tag{80}$$

Thus

$$\lambda_1 = -\frac{1}{N-m} \sum_{\substack{j \\ p_j \neq 0, l_j \neq 0}} f_{z_j}(\vec{z}) \tag{81-1}$$

$$\lambda_2 = -\lambda_1 - f_{z_{j_1}}(\vec{z}) \tag{81-2}$$

$$\lambda_3 = -\lambda_1 - f_{z_{j_2}}(\vec{z}) \tag{81-3}$$

$$\vdots$$

$$\lambda_{m+1} = -\lambda_1 - f_{z_{j_m}}(\vec{z}) \tag{81-4}$$

Substituting this latest result into Equation (71) yields

$$F_{\vec{z}}(\vec{z},\vec{\lambda}) = \begin{bmatrix} F_{z_1}(\vec{z},\vec{\lambda}) \\ F_{z_2}(\vec{z},\vec{\lambda}) \\ \vdots \\ F_{z_{j_1}}(\vec{z},\vec{\lambda}) \\ \vdots \\ F_{z_n}(\vec{z},\vec{\lambda}) \\ \vdots \\ F_{z_{j_2}}(\vec{z},\vec{\lambda}) \\ \vdots \\ F_{z_{j_m}}(\vec{z},\vec{\lambda}) \\ \vdots \\ F_{z_N}(\vec{z},\vec{\lambda}) \end{bmatrix} = \begin{bmatrix} -(f_{z_1}(\vec{z}) + \lambda_1) \\ -(f_{z_2}(\vec{z}) + \lambda_1) \\ \vdots \\ 0 \\ \vdots \\ -(f_{z_n}(\vec{z}) + \lambda_1) \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \\ -(f_{z_N}(\vec{z}) + \lambda_1) \end{bmatrix} \tag{82}$$

In compact notation

$$F_{z_j}(\vec{z},\vec{\lambda}) = \begin{cases} -(f_{z_j}(\vec{z}) + \lambda_1) & \text{if } l_j \notin I_0 \text{ and } p_j \notin I_0 \\ 0 & \text{if } l_j \in I_0 \text{ or } p_j \in I_0 \end{cases} \tag{83}$$

This completes the modifications to the algorithm.

SUMMARY OF THE NEW ALGORITHM

After computing the matrix A and the vector $\vec{b}$, the new steps are:

Step 1. Select a nominal point $\vec{z}$ such that

$$\vec{h}(\vec{z}) = C^T\vec{z} + \vec{g} = \vec{0} \tag{84}$$

Step 2. Compute

$$\vec{f}_{\vec{z}}(\vec{z}) = A\vec{z} + \vec{b} \tag{85}$$

Step 3. Compute

$$\lambda_1 = -\frac{1}{N-m} \sum_{\substack{j \\ p_j \notin I_0, l_j \notin I_0}} f_{z_j}(\vec{z}) \tag{86}$$

Step 4. Compute $F_{\vec{z}}(\vec{z}, \vec{\lambda})$ according to

$$F_{z_j}(\vec{z},\vec{\lambda}) = \begin{cases} -(f_{z_j}(\vec{z}) + \lambda_1) & \text{if } l_j \notin I_0 \text{ and } p_j \notin I_0 \\ 0 & \text{if } l_j \in I_0 \text{ or } p_j \in I_0 \end{cases} \tag{87}$$

Step 5. Compute

$$Q(\vec{z},\vec{\lambda}) = \vec{F}_{\vec{z}}^T(\vec{z},\vec{\lambda})\vec{F}_{\vec{z}}(\vec{z},\vec{\lambda}) \tag{88}$$

Step 6. Check for convergence. Is

$$\sqrt{Q(\vec{z},\vec{\lambda})/\vec{z}^T\vec{z}} \leq \epsilon \tag{89}$$

where ϵ is some small, preselected number. If (89) is satisfied, stop; if not, proceed with step 7.

Step 7. Is the iteration number equal to N-q? If so, stop; if not, proceed with step 8.

Step 8. Compute the directional coefficient β. If this is the first iteration, $\beta = 0$; otherwise

$$\beta = Q(\vec{z},\vec{\lambda})/Q(\hat{z},\hat{\lambda}) \tag{90}$$

Step 9. Determine the search direction

$$\vec{p} = -F_{\vec{z}}(\vec{z},\vec{\lambda}) + \beta\vec{p} \tag{91}$$

Step 10. Compute the optimum stepsize

$$\alpha = Q(\vec{z},\vec{\lambda})/\vec{p}^{T}A\vec{p} \tag{92}$$

Step 11. Compute the varied point

$$\vec{z}' = \vec{z} + \alpha\vec{p} \tag{93}$$

Step 12. Replace the old nominal values $\vec{z}$, $\vec{p}$, and $Q(\vec{z}, \vec{\lambda})$ with the new nominal values $\vec{z}'$, $\vec{p}$, and $Q(\vec{z}, \vec{\lambda})$

$$\vec{z} = \vec{z}' \tag{94-1}$$

$$\vec{p} = \vec{p} \tag{94-2}$$

$$Q(\hat{z},\hat{\lambda}) = Q(\vec{z},\vec{\lambda}) \tag{94-3}$$

and proceed to step 2.

It should be noted that when I_0 is the null set, the new algorithm automatically reduces to the old algorithm.

SELECTION OF A NOMINAL VECTOR

In deriving our algorithm, we have assumed that the nominal point $\vec{z}$ satisfies Equation (84). This assumption necessitates that we start the algorithm with such a vector. Although any such vector will suffice, we might as well use our degrees of freedom wisely and select a nominal that not only satisfies (84), but also produces the desirable property that the initial discrete function y_{l_j}, y_{p_j}, $j = 1, \ldots, N$ is of minimum modulus.

To accomplish this, we minimize

$$J = \frac{1}{2} \sum_{j=1}^{N} (y_{l_j}^2 + y_{p_j}^2) \tag{95}$$

with respect to the vector $\vec{z}$, subject to Equation (84). Equations (84) and (95) can be rewritten as

$$J = \frac{1}{2} \sum_{j=1}^{N} \left[(y_{p_j} + \delta_{l_j})^2 + y_{p_j}^2 \right] \tag{96}$$

and

$$\sum_{j=1}^{N} y_{p_j} + \frac{1}{2} \sum_{j=1}^{N} \delta_{l_j} = 0 \tag{97-1}$$

$$y_{p_j} = -\delta_{l_j} \quad \text{for } l_j \epsilon I_0, p_j \notin I_0 \tag{97-2}$$

$$y_{p_j} = 0 \quad \text{for } p_j \epsilon I_0 \tag{97-3}$$

From Equations (97-2) and (97-3) it can be seen that we only need to perform our minimization over those y_{p_j} for which $l_j \notin I_o$ and $p_j \notin I_o$, subject to the single constraint (97-1). Taking the partial of the Lagrangian

$$L = \frac{1}{2} \sum_{j=1}^{N} \left[(y_{p_j} + \delta_{l_j})^2 + y_{p_j}^2 \right] + \lambda \left(\sum_{j=1}^{N} y_{p_j} + \frac{1}{2} \sum_{j=1}^{N} \delta_{l_j} \right) \tag{98}$$

with respect to y_{p_j}, $l_j \notin I_o$, $p_j \notin I_o$ and setting it equal to zero produces

$$\frac{\partial L}{\partial p_j} = 2y_{p_j} + \lambda_{l_j} + \mu = 0 \qquad j = 1,\ldots,N \quad p_j, l_j \notin I_0 \tag{99-1}$$

or

$$-\frac{\lambda}{2} = y_{p_j} + \frac{1}{2}\lambda_{l_j} \qquad j = 1,\ldots,N \quad p_j, l_j \notin I_0 \tag{99-2}$$

while Equation (97-1) can be rearranged as

$$\sum_{j=1}^{N} (y_{p_j} + \frac{1}{2}\lambda_{l_j}) = 0 \tag{100}$$

Substituting (99-2) into (100) results in

$$\sum_{\substack{j \\ p_j \in I_0}} (y_{p_j} + \frac{1}{2}\lambda_{l_j}) + \sum_{\substack{j \\ l_j \in I_0}} (y_{p_j} + \frac{1}{2}\lambda_{l_j}) = \frac{(N-m)}{2}\lambda \tag{101}$$

Using Equations (97-2) and (97-3) we have

$$\lambda = \frac{(N-m)}{2}\lambda \tag{102}$$

where

$$\lambda = \frac{1}{2}\sum_{\substack{j \\ p_j \in I_0}} \lambda_{l_j} - \frac{1}{2}\sum_{\substack{j \\ l_j \in I_0}} \lambda_{l_j} \tag{103}$$

Placing this result into (99-2) yields

$$y_{p_j} = -\frac{\lambda_{l_j}}{2} + \frac{\lambda}{N-m} \qquad j = 1,\ldots,N \quad p_j, l_j \notin I_0 \tag{104}$$

Thus, our optimal nominal $z_j = y_{p_j}$, $j = 1, \ldots, N$ is defined via Equations (97-2), (97-3), (103) and (104).

When I_o is empty, Equation (104) reduces to

$$y_{p_j} = -\frac{\delta_{1_j}}{2} \qquad j = 1, \ldots, N \qquad (105)$$

as one would expect.

NUMERICAL EXAMPLES

Test Problems

The conjugate gradient-projection algorithm was developed into a computer-programmed subroutine written in FORTRAN V. The algorithm was then applied to several test problems. In each case, the nominal function $\vec{z}$ was selected via Equations (97-2), (97-3), (103) and (104). Convergence was defined by

$$\sqrt{Q(\vec{z},\vec{\lambda})/\vec{z}^T\vec{z}} \leq 10^{-6} \tag{106}$$

and double precision arithmetic was used in the calculation of the optimum stepsize α and the directional coefficient β. The algorithm was terminated whenever Inequality (106) was satisfied or the number of iterations reached N-q.[11]

For these problems, the ephemeris errors were represented by various sinusoids. In each case, the observations δ_j, $j = 1, \ldots, N$ were corrupted by Gaussian noise with a sigma level of $0.01\sqrt{2}$ (1% of the maximum ephemeris error) and the average number of observations contained in the shortest wavelength of the sinusoid was recorded.

Three different weighting techniques in combination with three different spacing schemes were used. The weighting techniques were:

(1) Equal weighting in which all of the weights w_i, $i = 1, \ldots, 2N-1$ are equal.

(2) Inverse weighting in which $w_i = \dfrac{\xi}{t_{i+1}-t_i}$, $i = 1, \ldots, 2N-1$

where ξ is a scaling factor and

(3) Inverse square weighting in which $w_i = \dfrac{\xi}{(t_{i+1}-t_i)^2}$, $i = 1, \ldots, 2N-1$ where again ξ is a scaling factor.

The spacing schemes used are given below.

---

[11] Theoretically, the algorithm has to converge in at most N-q iterations. However, due to round-off errors, especially in the calculation of the A-orthogonal directions, it is possible to impose a convergence criterion that is tighter than the algorithm is capable of achieving.

Table 1. Spacing Schemes

Scaled Time		
Scheme 1	Scheme 2	Scheme 3
.3	.1	.20
.4	.2	.25
.5	.5	.30
.6	.8	.35
.7	.9	.40
1.3	1.1	.45
1.4	1.2	.50
1.5	1.5	.55
1.6	1.8	.60
1.7	1.9	.65
2.3	2.1	2.20
2.4	2.2	2.25
2.5	2.5	2.30
2.6	2.8	2.35
2.7	2.9	2.40
3.3	3.1	2.45
3.4	3.2	2.50
3.5	3.5	2.55
3.6	3.8	2.60
3.7	3.9	2.65
.	.	4.20
.	.	4.25
.	.	4.30
.	.	.
.	.	.
.	.	.
.	.	.
		.

The ephemeris errors were as follows:

Problem 1:
$$f(t) = \sin \tfrac{\pi}{2} t \tag{107}$$

Problem 2:
$$f(t) = \tfrac{1}{2}[\sin \tfrac{\pi}{2} t + \sin \tfrac{\pi}{4} t] \tag{108}$$

Problem 3:
$$f(t) = \sin \tfrac{\pi}{5} t \tag{109}$$

Problem 4:
$$f(t) = \sin \left(\tfrac{\pi}{10} t - \tfrac{5}{4}\right) \tag{110}$$

Problem 5:
$$f(t) = \frac{1}{2}\left[\sin \frac{\pi}{2}t + \sin \pi t\right] \tag{111}$$

Problem 6:
$$f(t) = \begin{cases} 0 \text{ for } t < t \\ \sin \frac{\pi}{2}t \text{ for } t \geq t \end{cases} \tag{112}$$

In each case, 25 intersections were considered. The ordered sets $\{p_j : j = 1, \ldots, 25\}$ and $\{I_j : j = 1, \ldots, 25\}$ remained unchanged with

$$\{p_j : j = 1, \ldots, 25\} = \{3,5,8,1,2,6,4,7,9,12,17,18,11,14,22, 25,24,23,26,29,31,32,35,34,39\} \tag{113}$$

and

$$\{I_j : j = 1, \ldots, 25\} = \{10,13,15,16,19,20,21,27,28,30,33,36,37, 38,40,41,42,43,44,45,46,47,48,49,50\} \tag{114}$$

In Problems 1-5, the set I_0 was empty. In Problem 6, it was assumed that the track had previously been corrected from time $t = 0$ to $t = t_{25}$ and that there was a huge time gap between t_{25} and t_{26}. These conditions were simulated by defining the ephemeris error via Equation (112), setting the weight w_{25} equal to zero, and defining $I_0 = \{i : i = 1, \ldots, 25\}$.

The results for these problems are illustrated in Figures 1-18. Problem 1 is associated with Figures 1-3, Problem 2 with Figures 4-6, Problem 3 with Figures 7-9, etc. In most cases the conjugate gradient-projection algorithm required less than the maximum number of 24 iterations to converge. The plotted solutions represent the sum of the unbiased estimate obtained from the algorithm plus the bias in the true ephemeris error. Note that as the average number of observations per shortest wavelength increases, so does the accuracy of the solution. At 50 observations/wavelength, the accuracy is down to the noise level of the system. In some cases, depending on the spacing scheme, this level can be achieved with 25 observations/wavelength. Even with as few as 5 observations/wavelength the ephemeris errors can still be substantially reduced. At this point, however, relative to our sampling rate, the ephemeris errors begin to look like high-frequency errors. Since the conjugate gradient-projection algorithm produces a low-frequency result, the generated solution may be of lower frequency than the true one, as evidenced in Figure 14.

Not to be overlooked is the fact that the algorithm is relatively insensitive to the weighting scheme employed. Furthermore, this insensitivity increases as the number of observations per wavelength increases. Because of its performance over a range of observation frequencies, inverse weighting was deemed to be superior to inverse-square weighting.

SEASAT Geoid Height Data

Moving from "Alice in Wonderland" into the real world, an analysis of the first 393 revolutions of SEASAT was conducted. From this initial segment of the satellite's trek, good geoid height information was obtained during 125 of the revolutions. From this latter group of revolutions, 2564 ground-track intersections containing data were produced. The conjugate gradient-projection algorithm was then applied to the resulting 2564 geoid height differences. Since time information had been previously removed from the data, inverse weighting based on revolution number and fractions thereof was employed. The algorithm required 840 iterations to converge.

A cubic spline with evenly-spaced nodes was then fitted in the least-square sense to the geoid-height-error output of the algorithm. The spline was represented as a linear combination of spline basis functions and a pseudo-inverse solution of coefficients was obtained via a singular value decomposition (SVD). Spurious oscillations were damped by zeroing out an appropriate number of the singular values. A transformation was performed prior to entering the SVD so that, whenever the solution was non-unique, the "minimum curvature"[12] solution was obtained (See Reference 5).

The results for revolutions 145 through 151, 207 through 213, 234 through 240, and 274 through 280 are given in Figures 19-22, respectively. For these segments, the rms error of the spline fit ranges between 17 and 20 centimeters.

In order to evaluate the accuracy of the geoid height correction, a fictitious geoid height error was assumed. As a function of the revolution number ν, this fictitious error was defined by

$$f(\nu) = \frac{3}{2}\left[\sin 3\pi\nu + \sin \frac{3}{2}\pi\right] \qquad (115)$$

This particular function was chosen because its frequency content is similar to that observed in Figures 19-22 — the highest frequency present being approximately $1\frac{1}{2}$ cycles per revolution.

After corrupting Equation (115) with Gaussian noise at a sigma level of 19 centimeters, fictitious geoid height differences were computed at the ground-track intersections. These differences were then fed into the conjugate gradient-projection algorithm; as before, a cubic spline was fitted to the geoid-height-error output of the algorithm. A comparison between this spline fit and the fictitious geoid height error is given in Figures 23-26. This comparison shows (for the revolution spans considered here) that over those segments of the satellite's track where no geoid height data exists (over land), we can obtain an ***extrapolated***[13] ***estimate of the altitude error to an rms-accuracy of 22 centimeters (RMS1). Over these segments containing geoid height data (over the ocean), we can obtain an interpolated estimate of the altitude error to an rms-accuracy of 15 centimeters (RMS2).***

[12] The L_2 norm of the second derivative was minimized.

[13] Interpolation between data sets.

Figure 1. 10 Observations/Wavelength

Spacing Scheme 1 Noise Sigma = .01

RMS = .707 RMS1 = .201

RMS2 = .111 RMS3 = .078

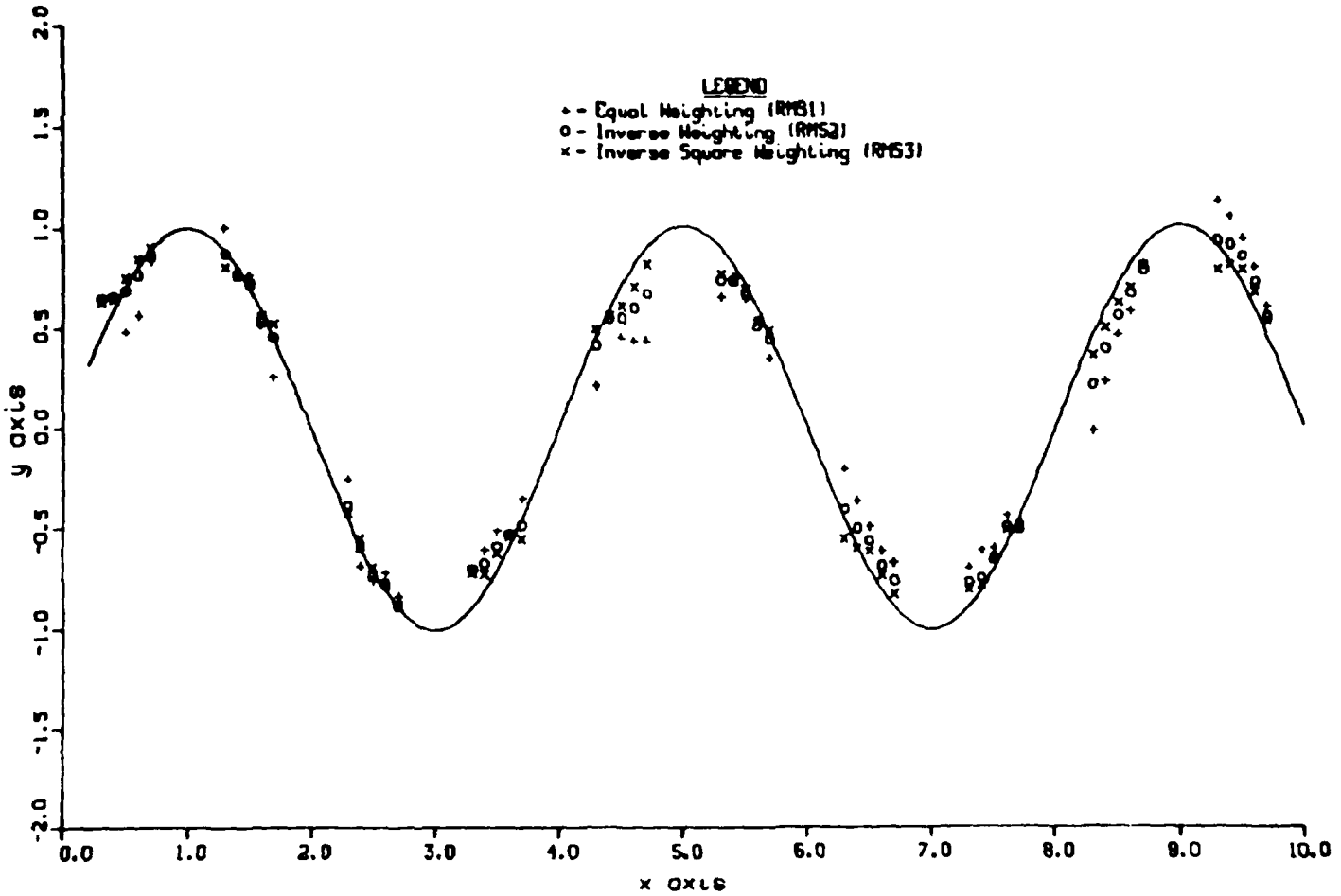


Figure 2. 10 Observations/Wavelength

Spacing Scheme 2 Noise Sigma = .01

RMS = .707 RMS1 = .158

RMS2 = .131 RMS3 = .121

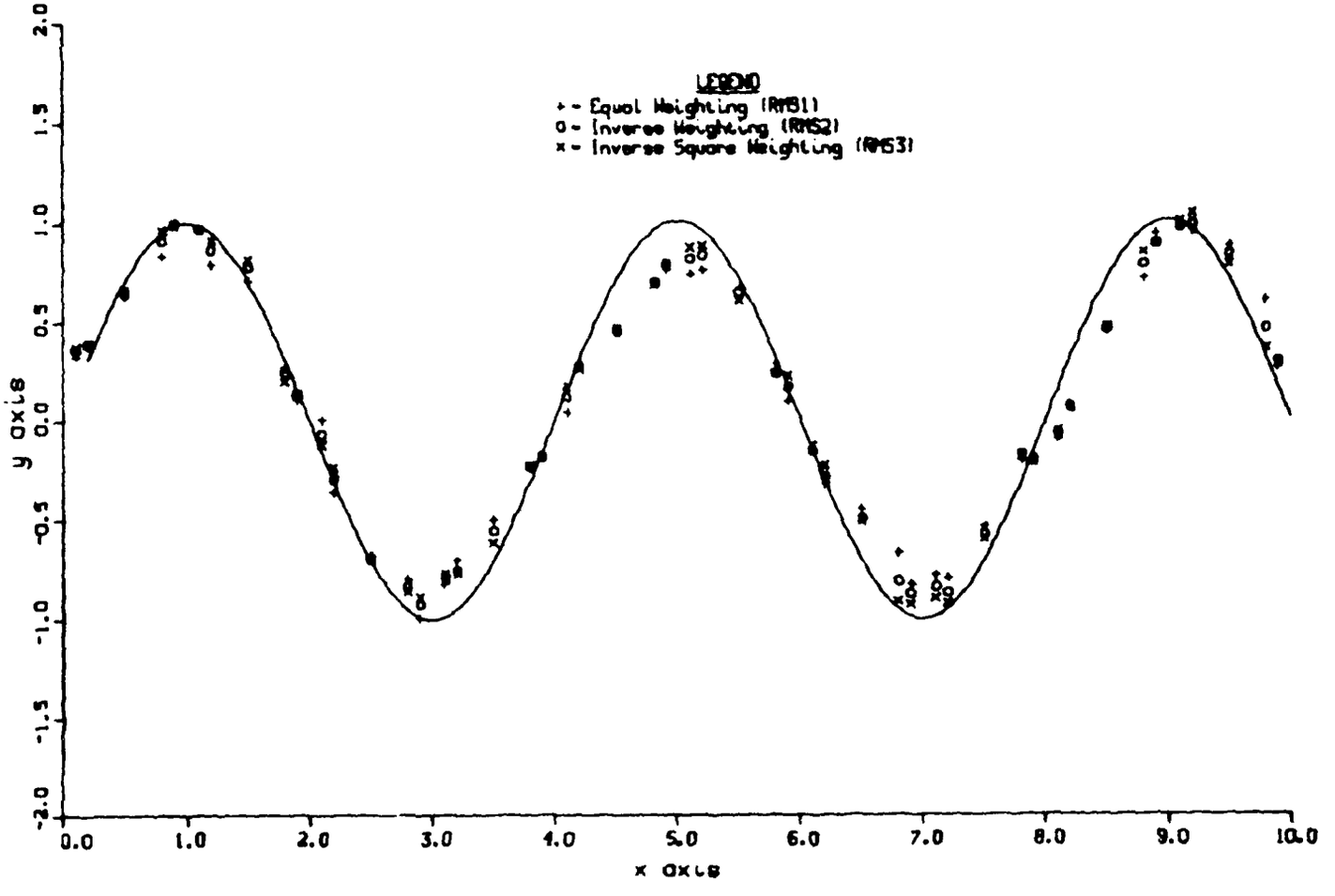


Figure 3. 10 Observations/Wavelength

Spacing Scheme 3 Noise Sigma - .01

RMS - .628 RMS1 - .202

RMS2 - .020 RMS3 - .028

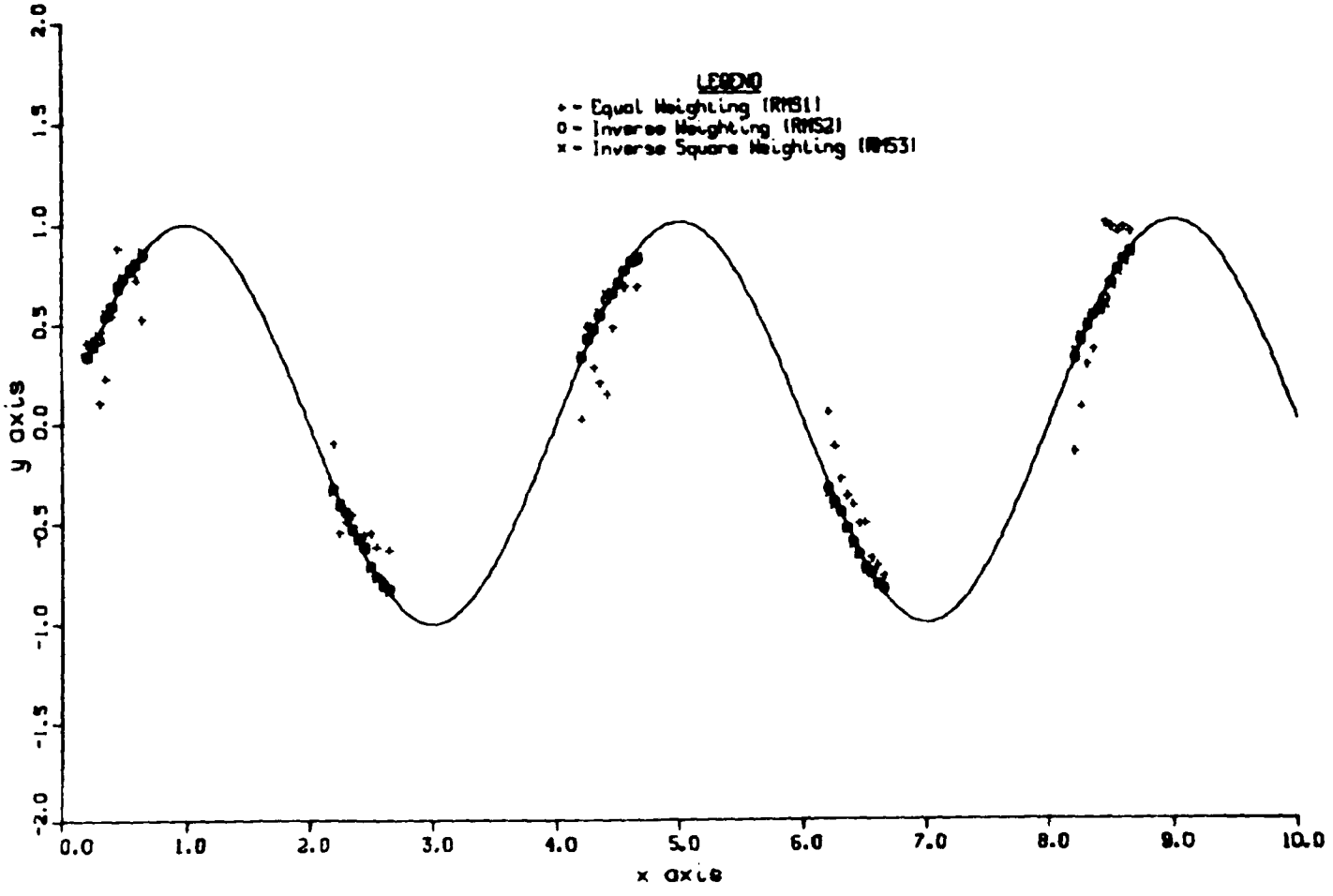


Figure 4. 10 Observations/Wavelength

Spacing Scheme 1 Noise Sigma = .01

RMS = .543 RMS1 = .121

RMS2 = .059 RMS3 = .038

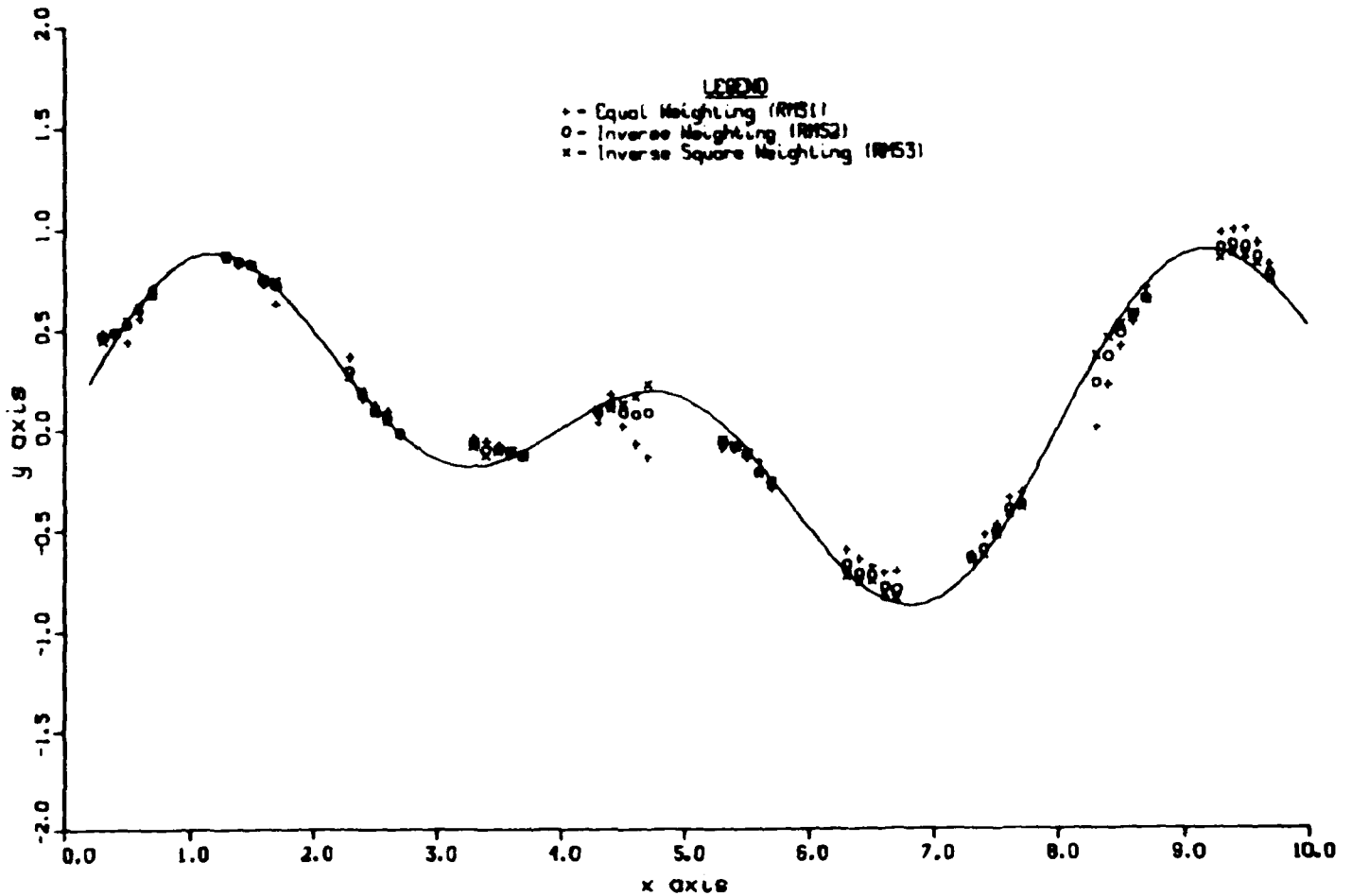


Figure 5. 10 Observations/Wavelength

Spacing Scheme 2 Noise Sigma = .01

RMS = .540 RMS1 = .078

RMS2 = .069 RMS3 = .072

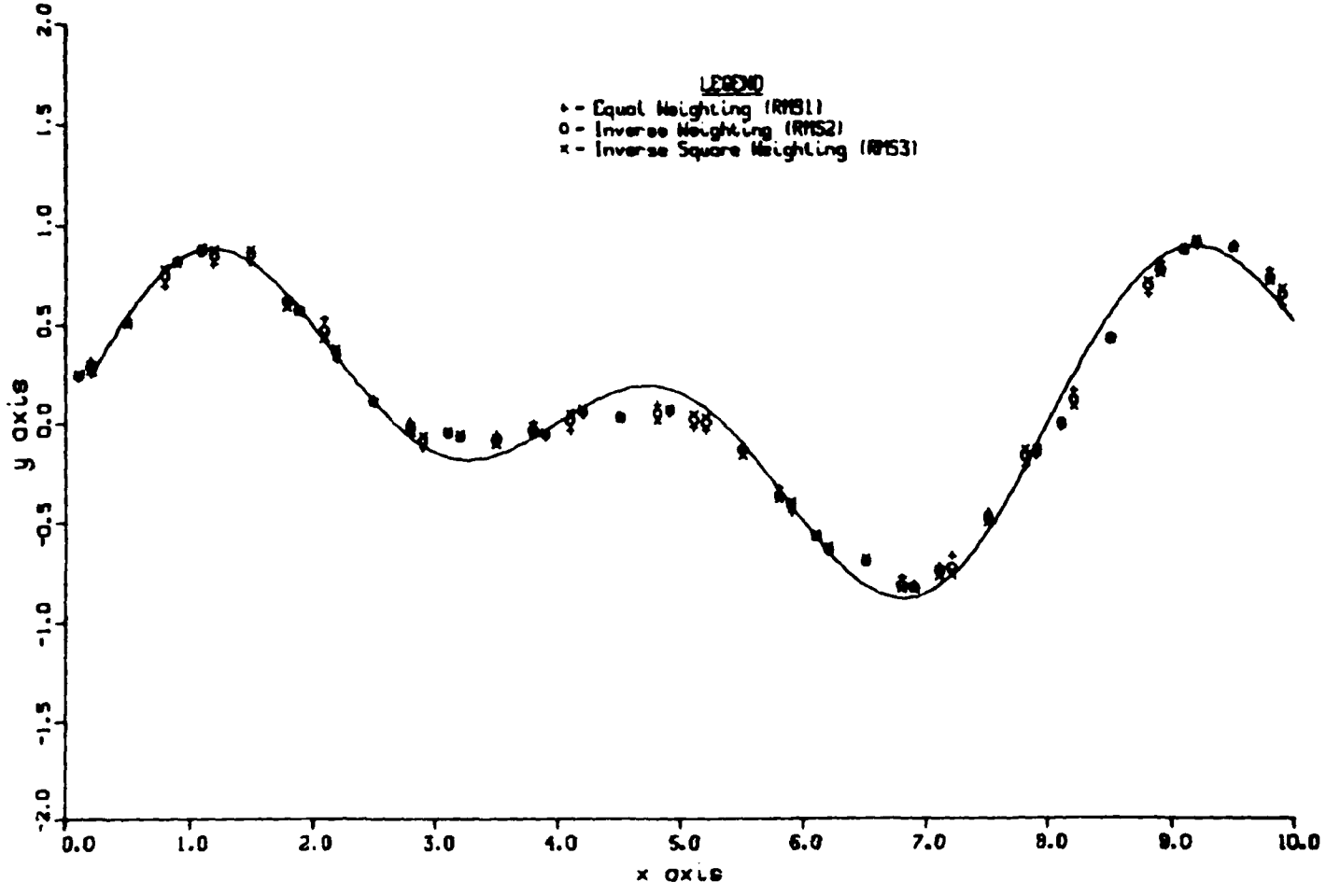


Figure 6. 10 Observations/Wavelength

Spacing Scheme 3 Noise Sigma = .01

RMS = .475 RMS1 = .133

RMS2 = .011 RMS3 = .015

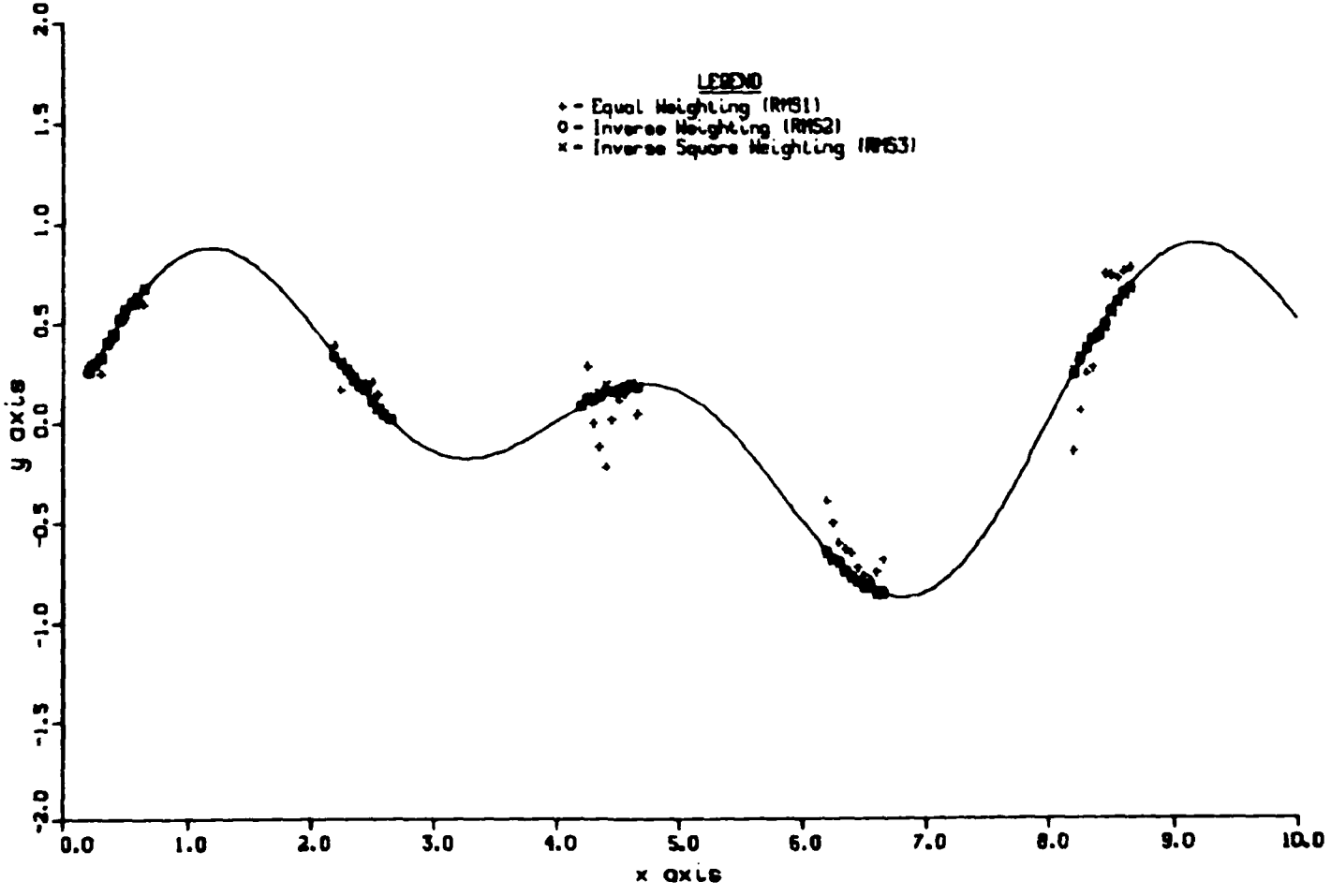


Figure 7. 25 Observations/Wavelength

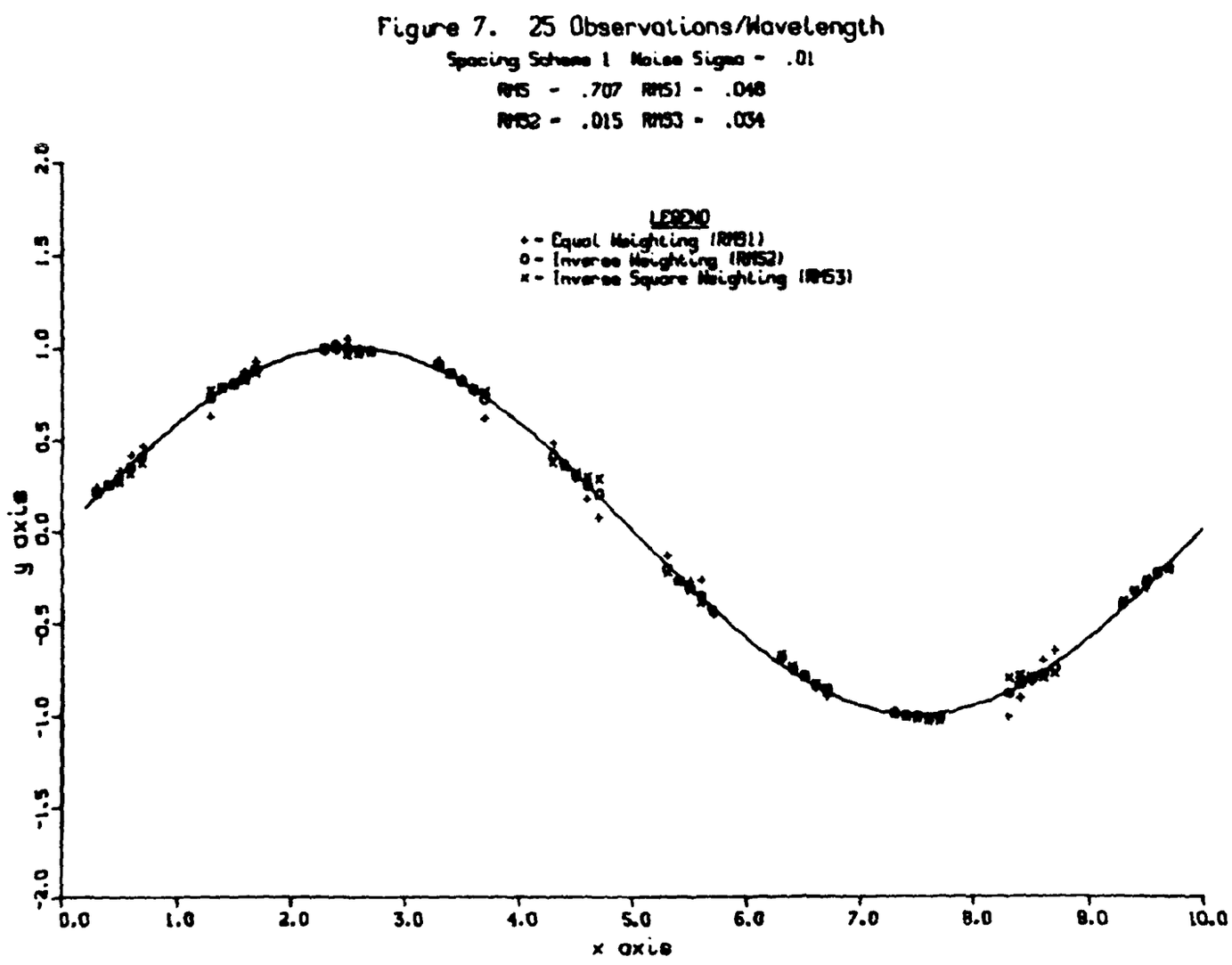


Figure 8. 25 Observations/Wavelength

Spacing Scheme 2 Noise Sigma = .01

RMS = .707 RMS1 = .030

RMS2 = .018 RMS3 = .024

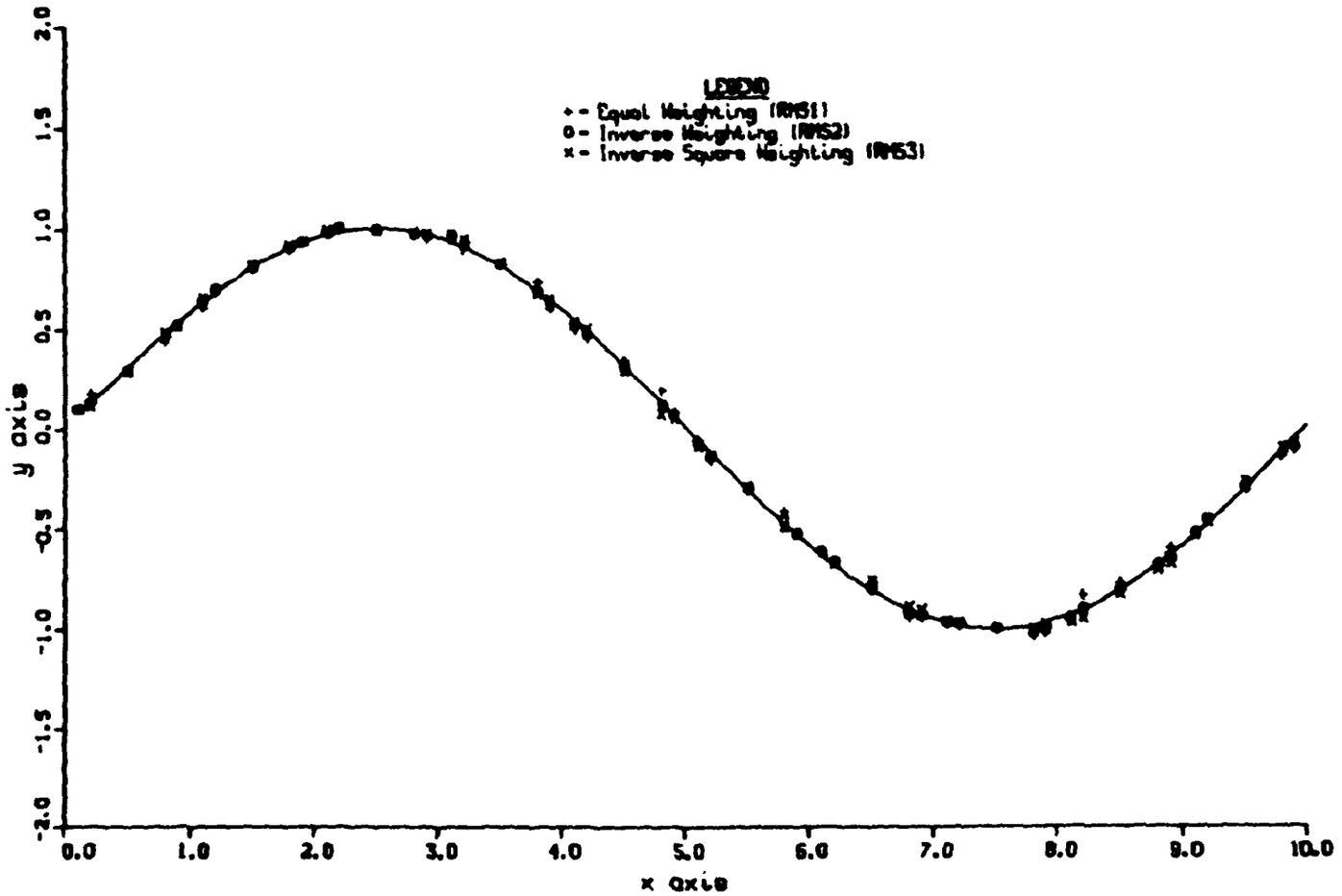


Figure 9. 25 Observations/Wavelength

Spacing Scheme 3 Noise Sigma = .01

RMS = .707 RMS1 = .094

RMS2 = .008 RMS3 = .011

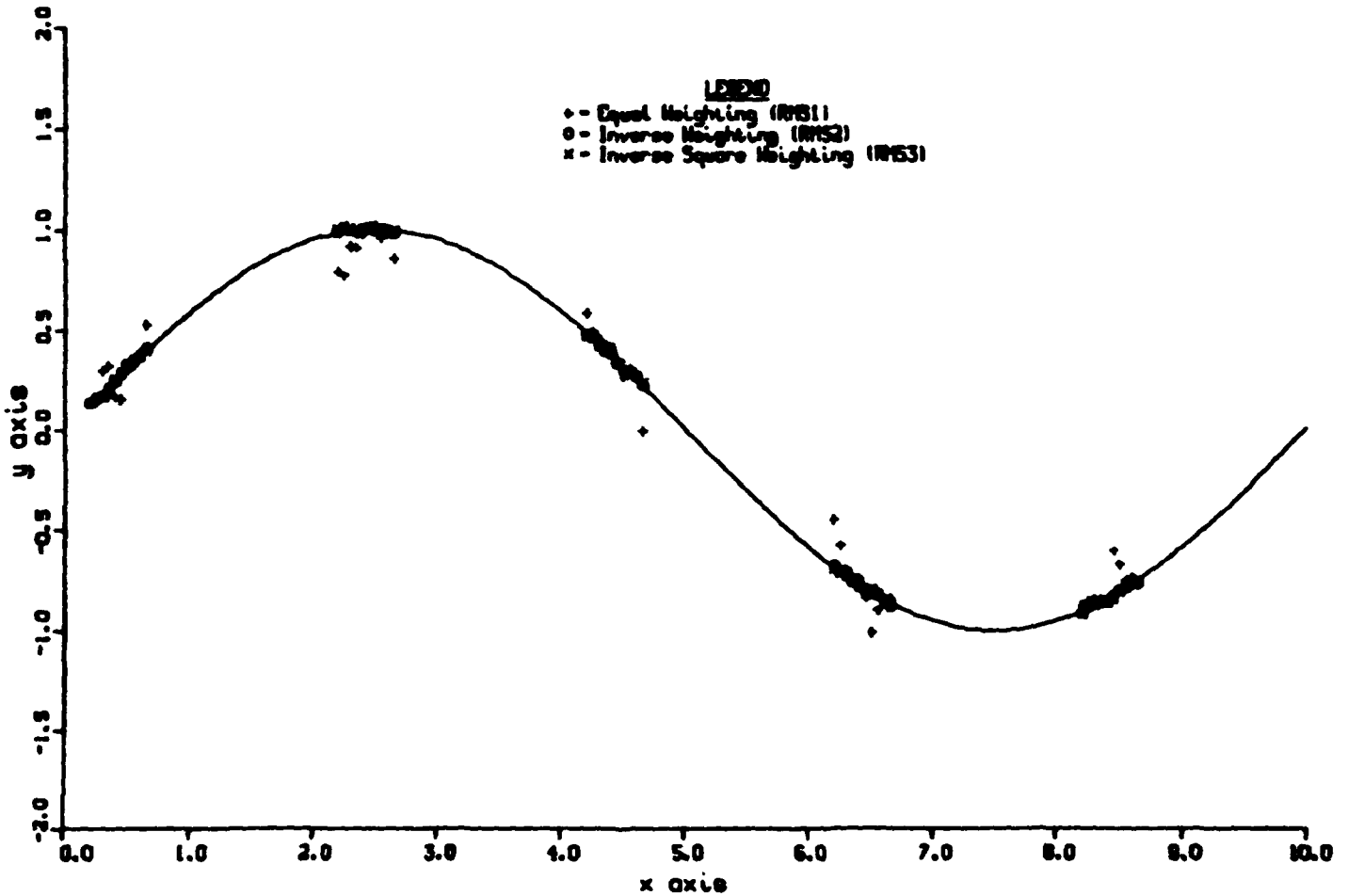
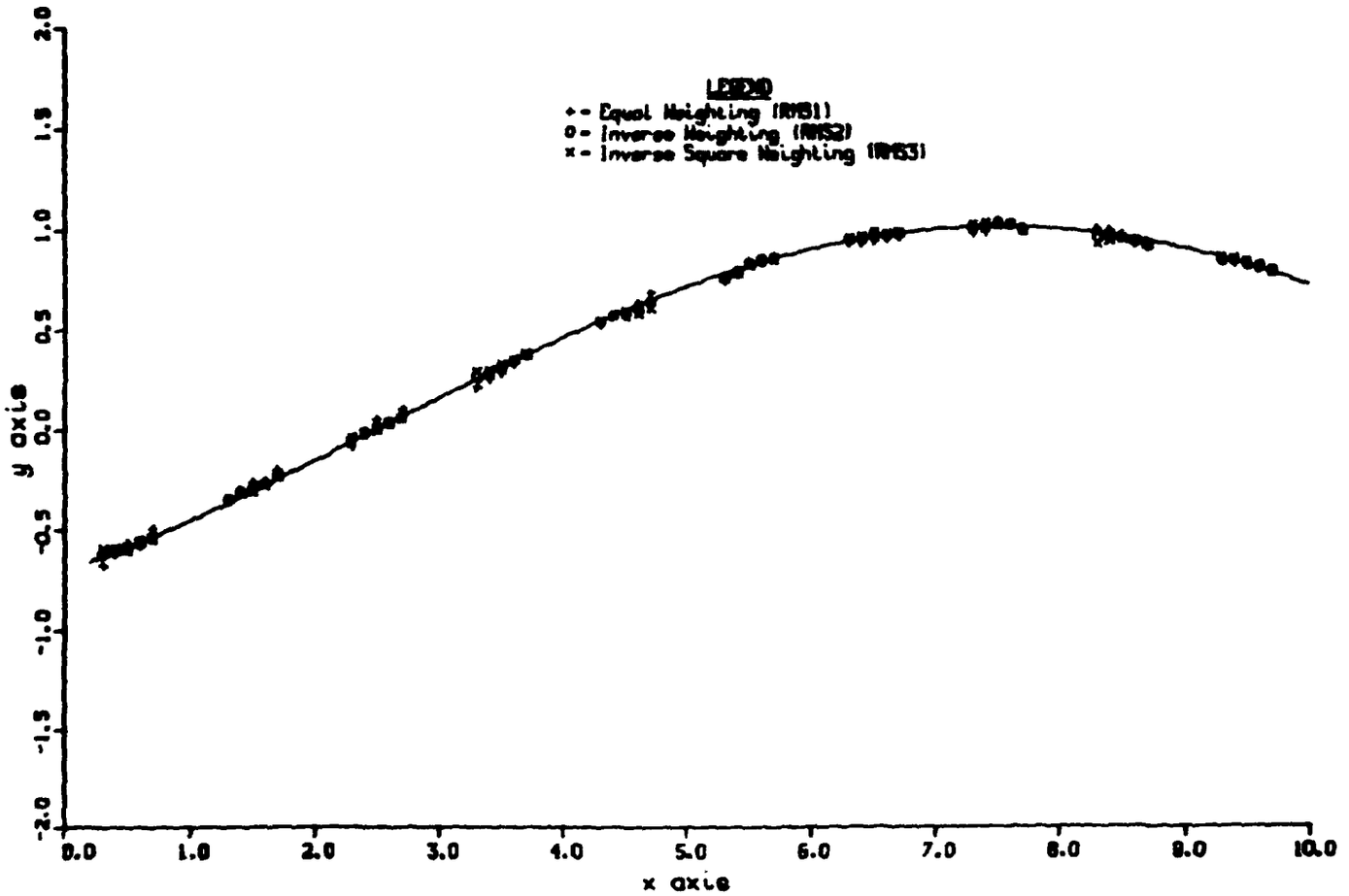


Figure 10. 50 Observations/WaveLength

Spacing Scheme 1 Noise Sigma = .01

RMS = .707 RMS1 = .024

RMS2 = .011 RMS3 = .018



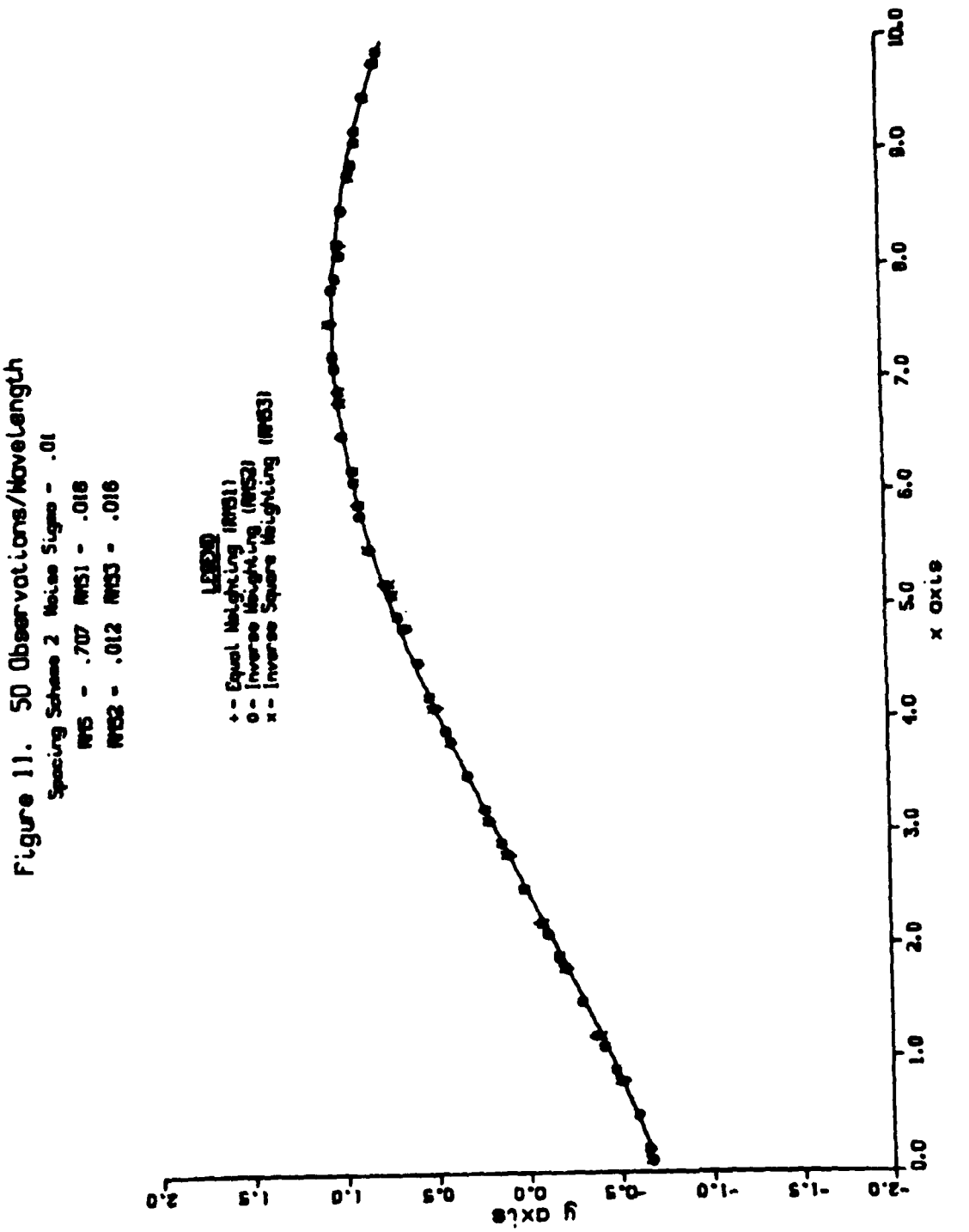


Figure 11. 50 Observations/Wavelength
Spacing Scheme 2 Noise Sigma = .01
RMS = .707 RMS1 = .018
RMS2 = .012 RMS3 = .016
LEGEND
+ = Equal Weighting (RMS1)
o = Inverse Weighting (RMS2)
x = Inverse Square Weighting (RMS3)
y axis
x axis
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
-2.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0

Figure 12. 50 Observations/Wavelength

Spacing Scheme 3 Noise Sigma = .01

RMS = .707 RMS1 = .054

RMS2 = .011 RMS3 = .010

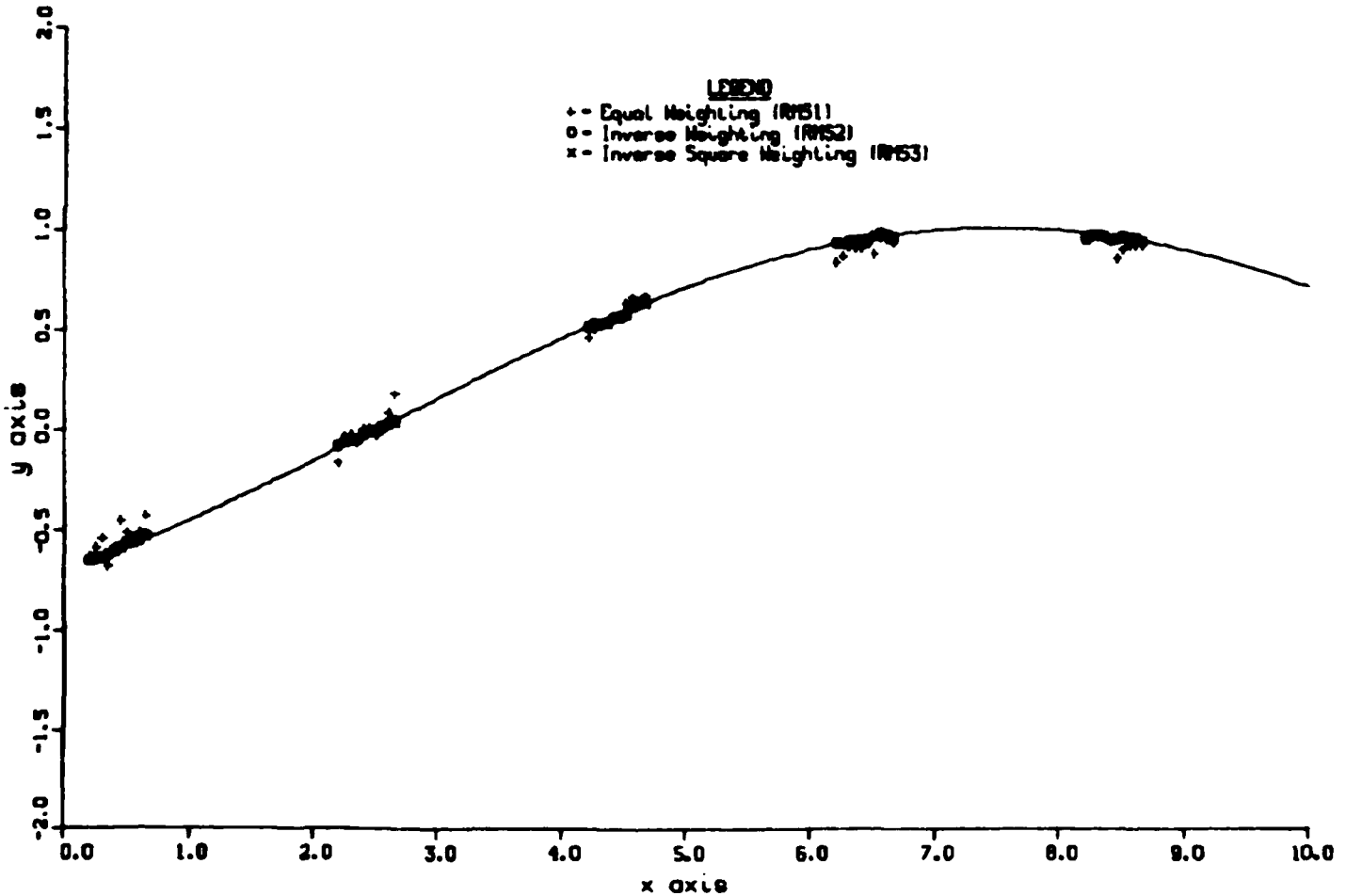


Figure 13. 5 Observations/Wavelength

Spacing Scheme 1 Noise Sigma = .01

RMS = .575 RMS1 = .339

RMS2 = .181 RMS3 = .084

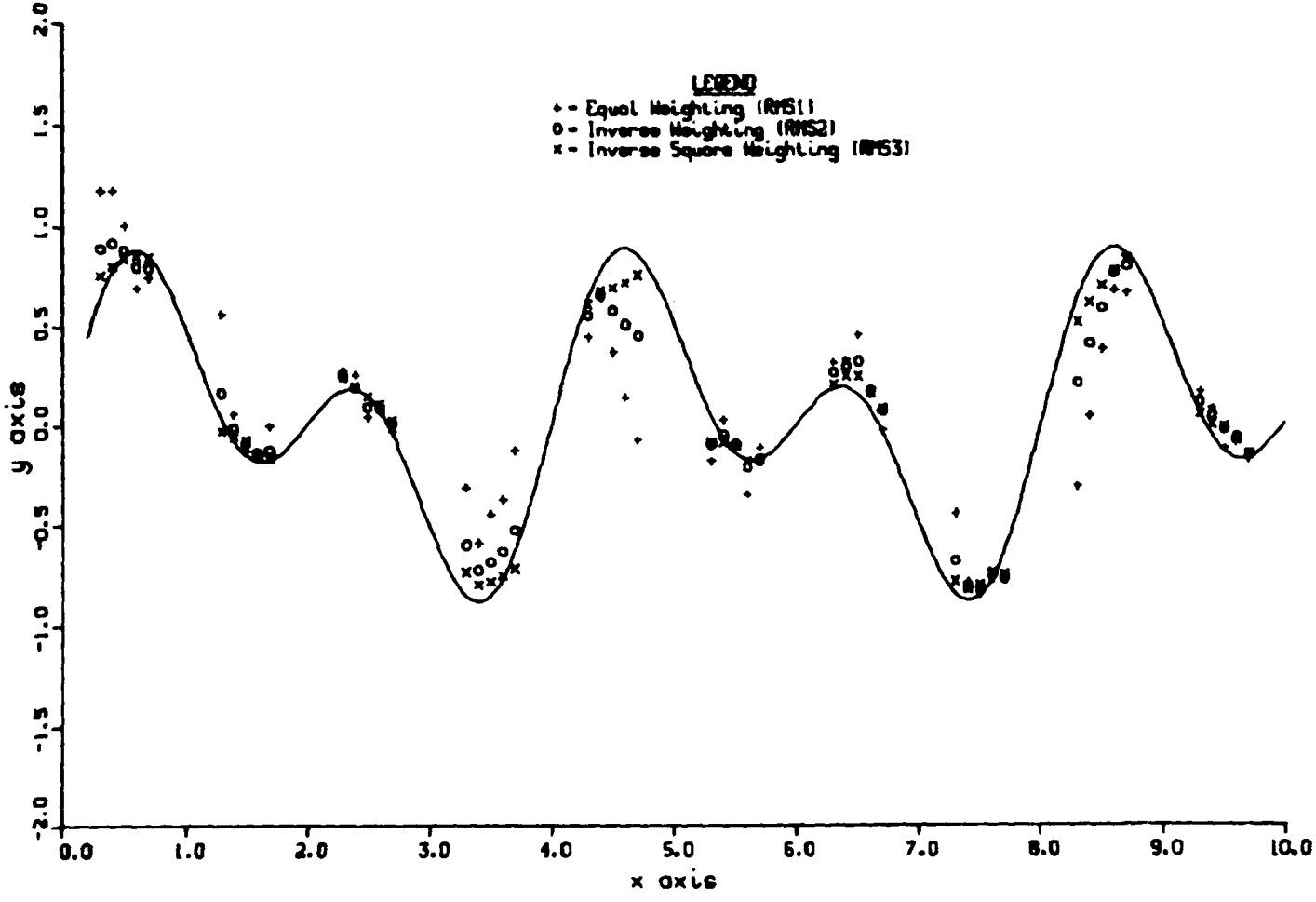


Figure 14. 5 Observations/Wavelength

Spacing Scheme 2 Noise Sigma = .01

RMS = .168 RMS1 = .256

RMS2 = .258 RMS3 = .288

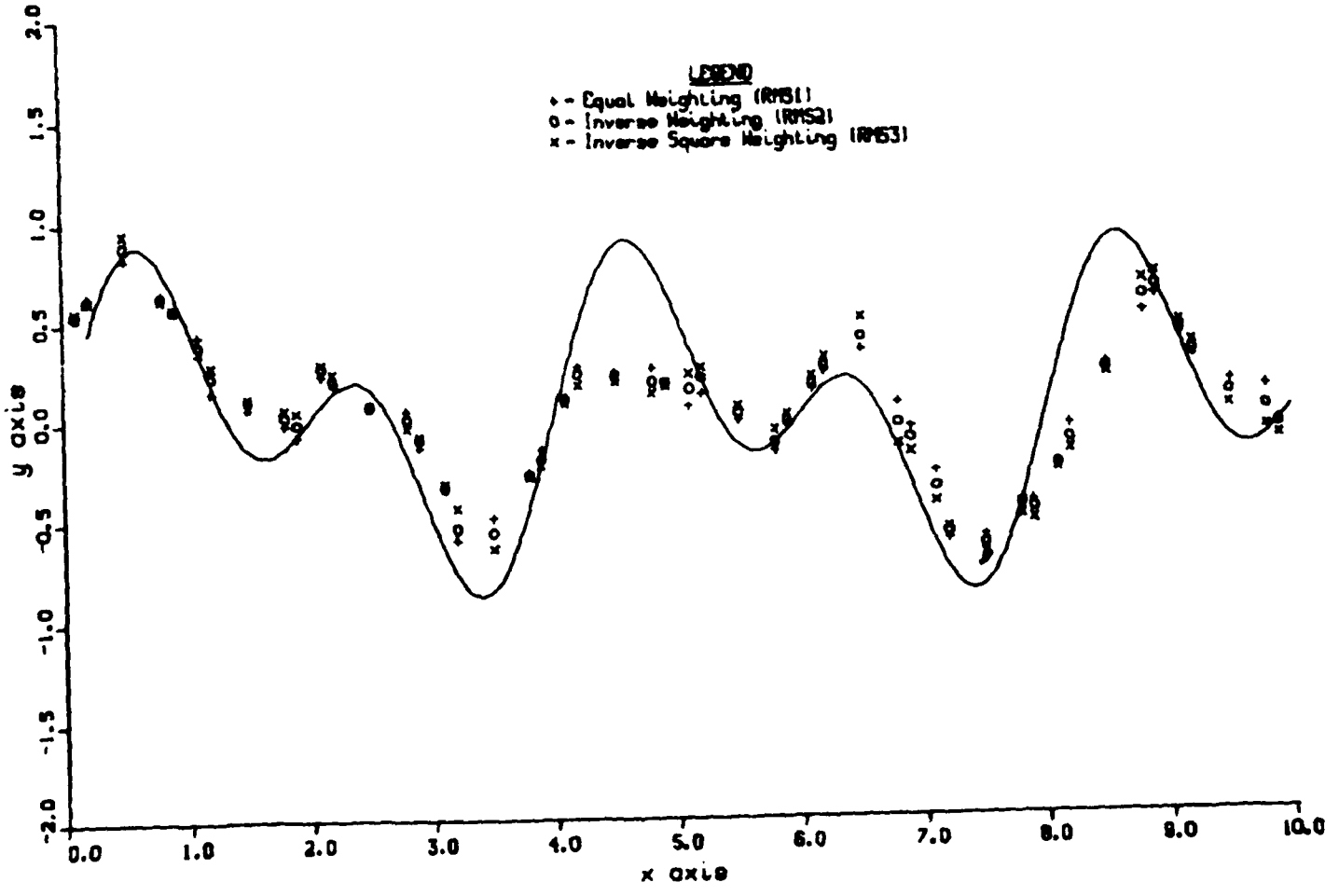


Figure 15. 5 Observations/Wavelength

Spacing Scheme 3 Noise Sigma = .01

RMS = .591 RMS1 = .100

RMS2 = .017 RMS3 = .021

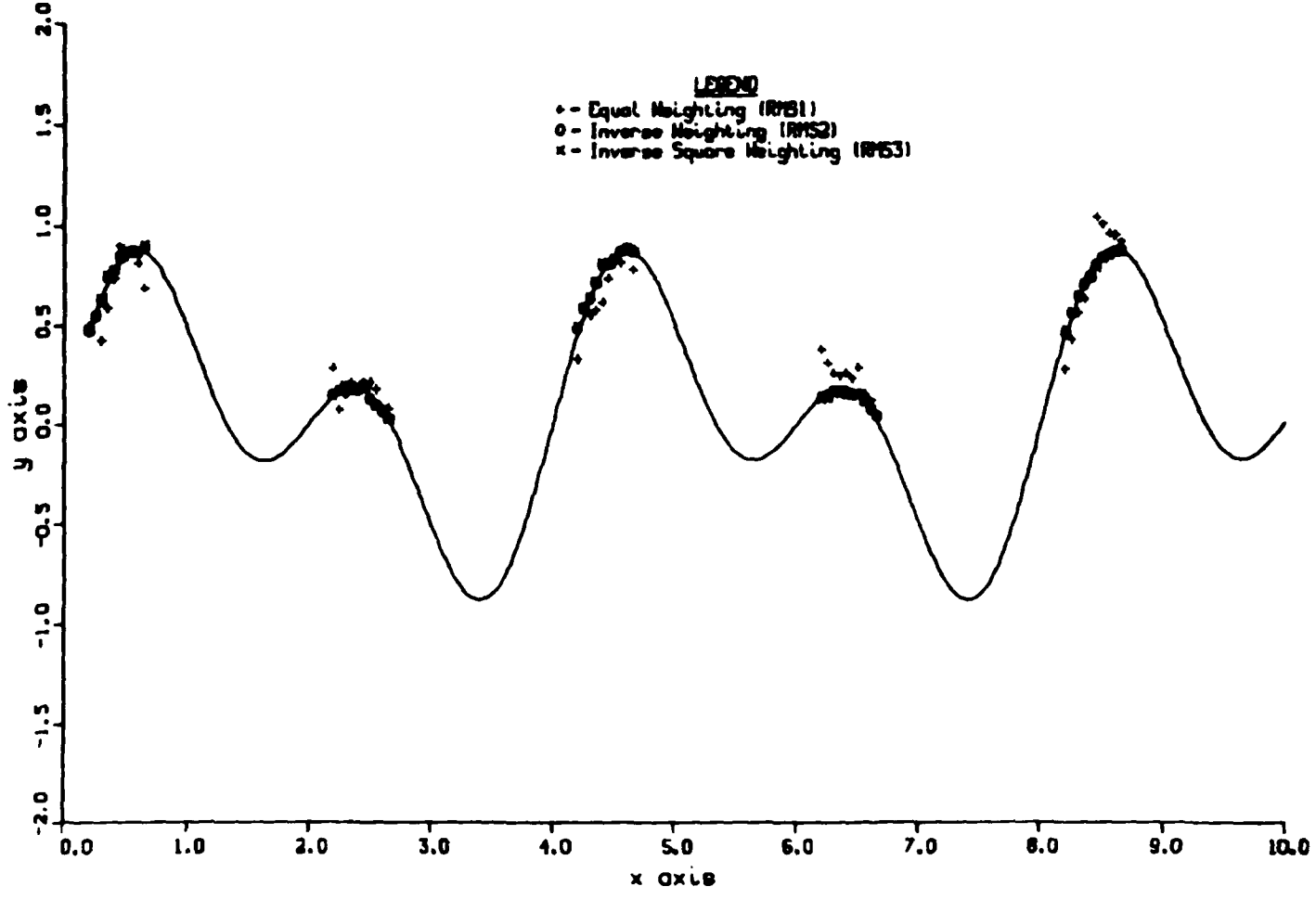


Figure 16. 10 Observations/Wavelength

Spacing Scheme 1 Noise Sigma = .01

RMS = .707 RMS1 = .127

RMS2 = .140 RMS3 = .161

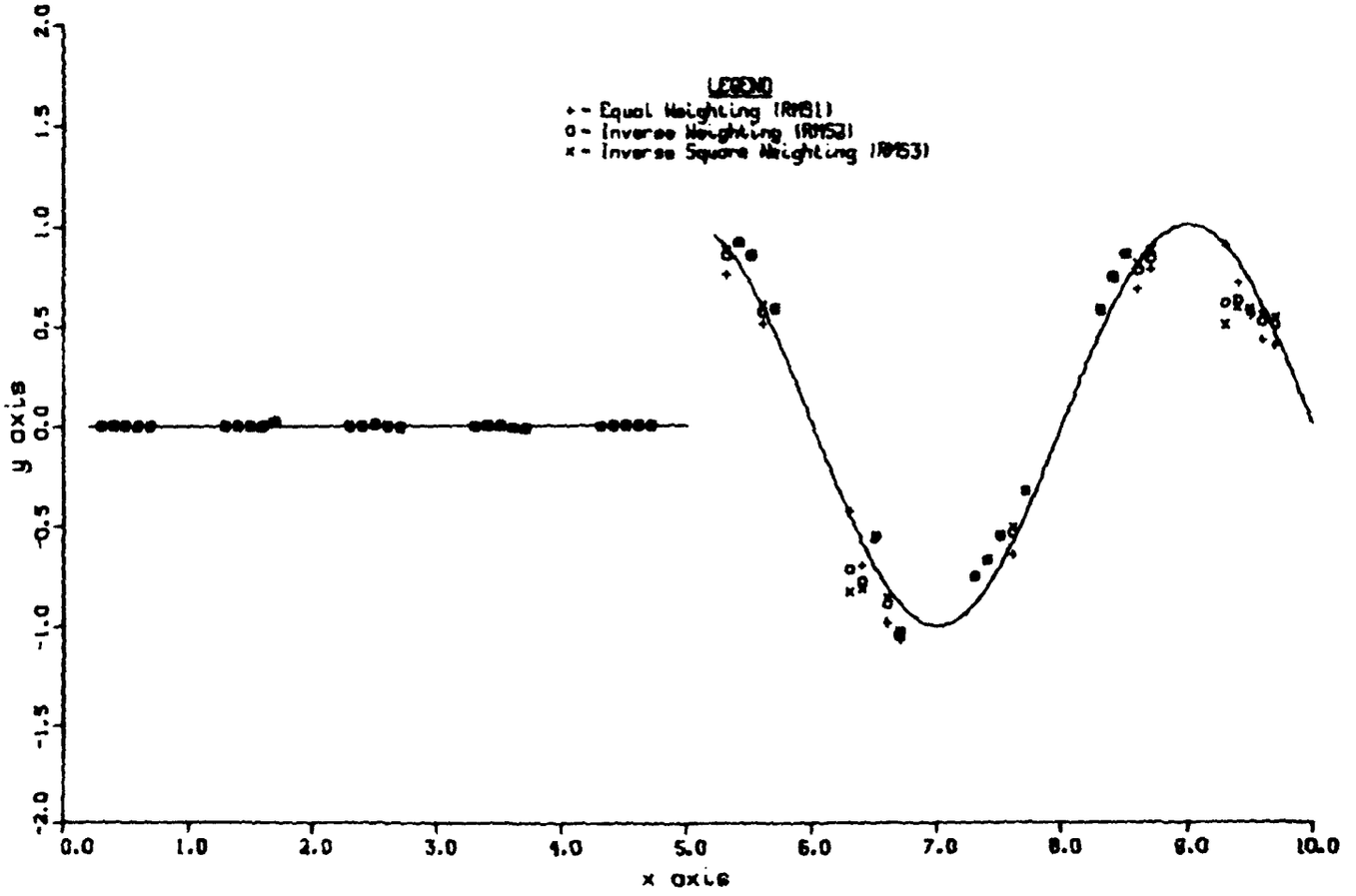


Figure 17. 10 Observations/Wavelength

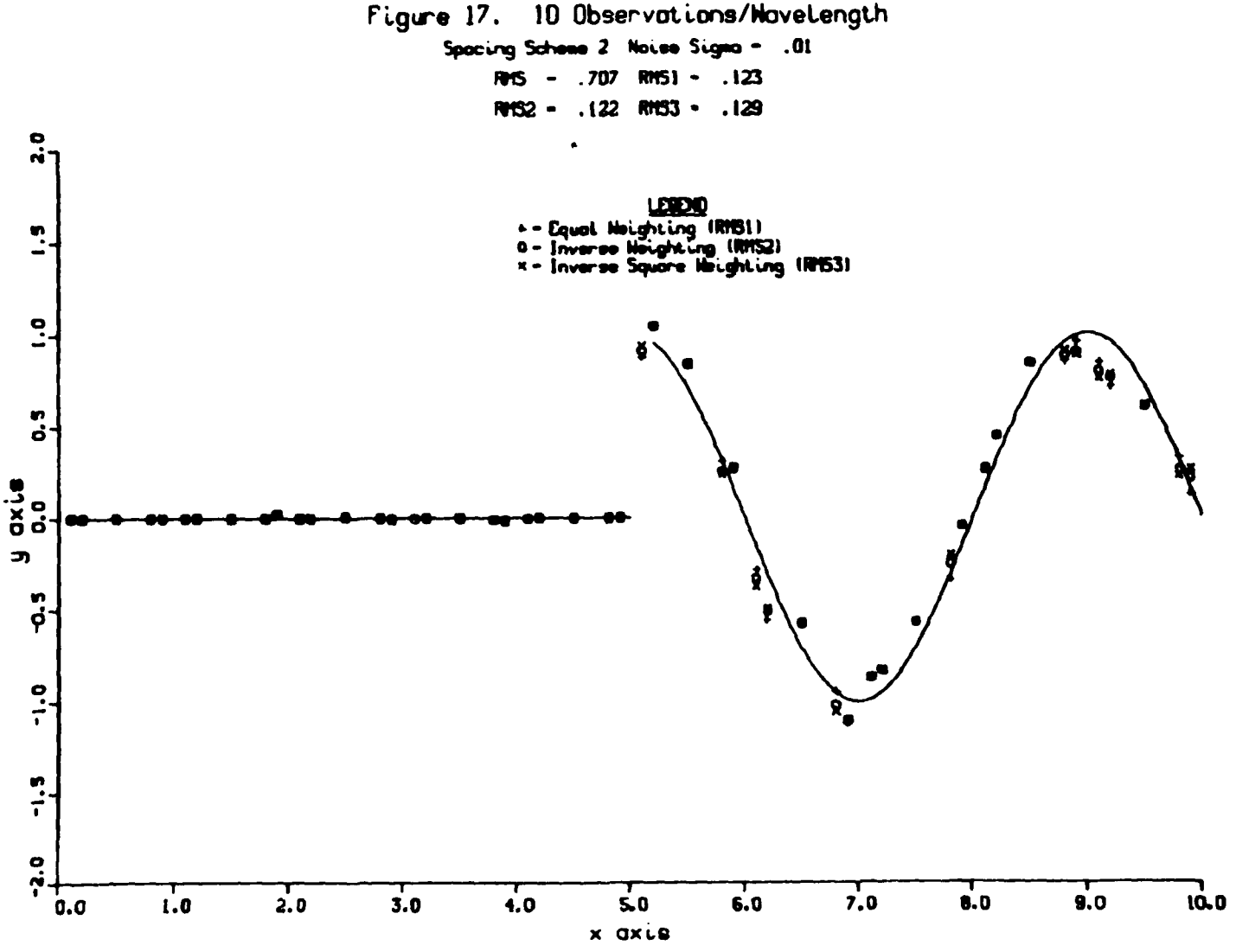


Figure 18. 10 Observations/Wavelength

Spacing Scheme 3 Noise Sigma = .01

RMS = .657 RMS1 = .157

RMS2 = .143 RMS3 = .145

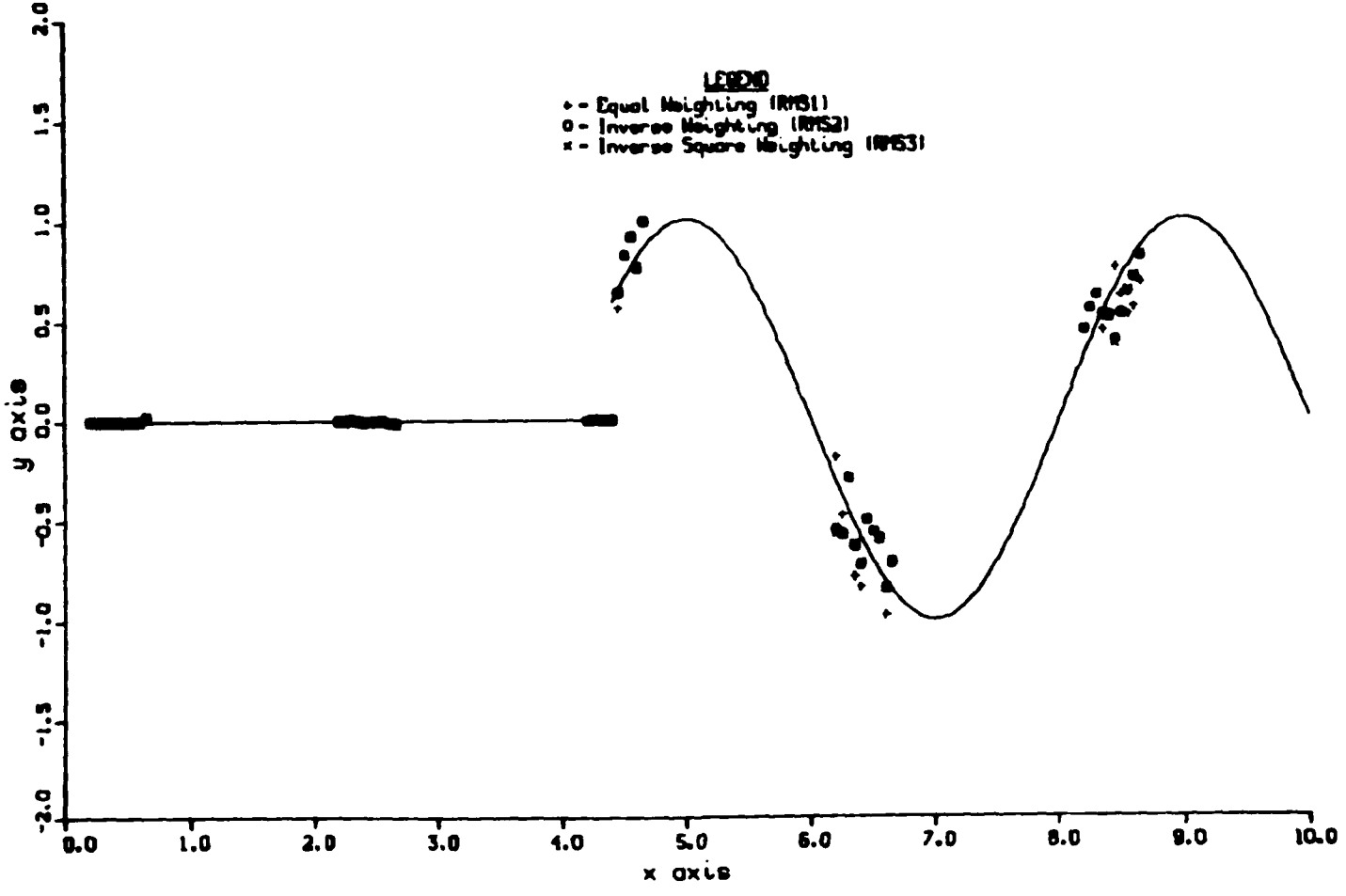


Figure 19. 7 Revolutions of SEASAT Data

Typical Spline Correction (RMS = .20)

(Pseudo-Inverse Solution)

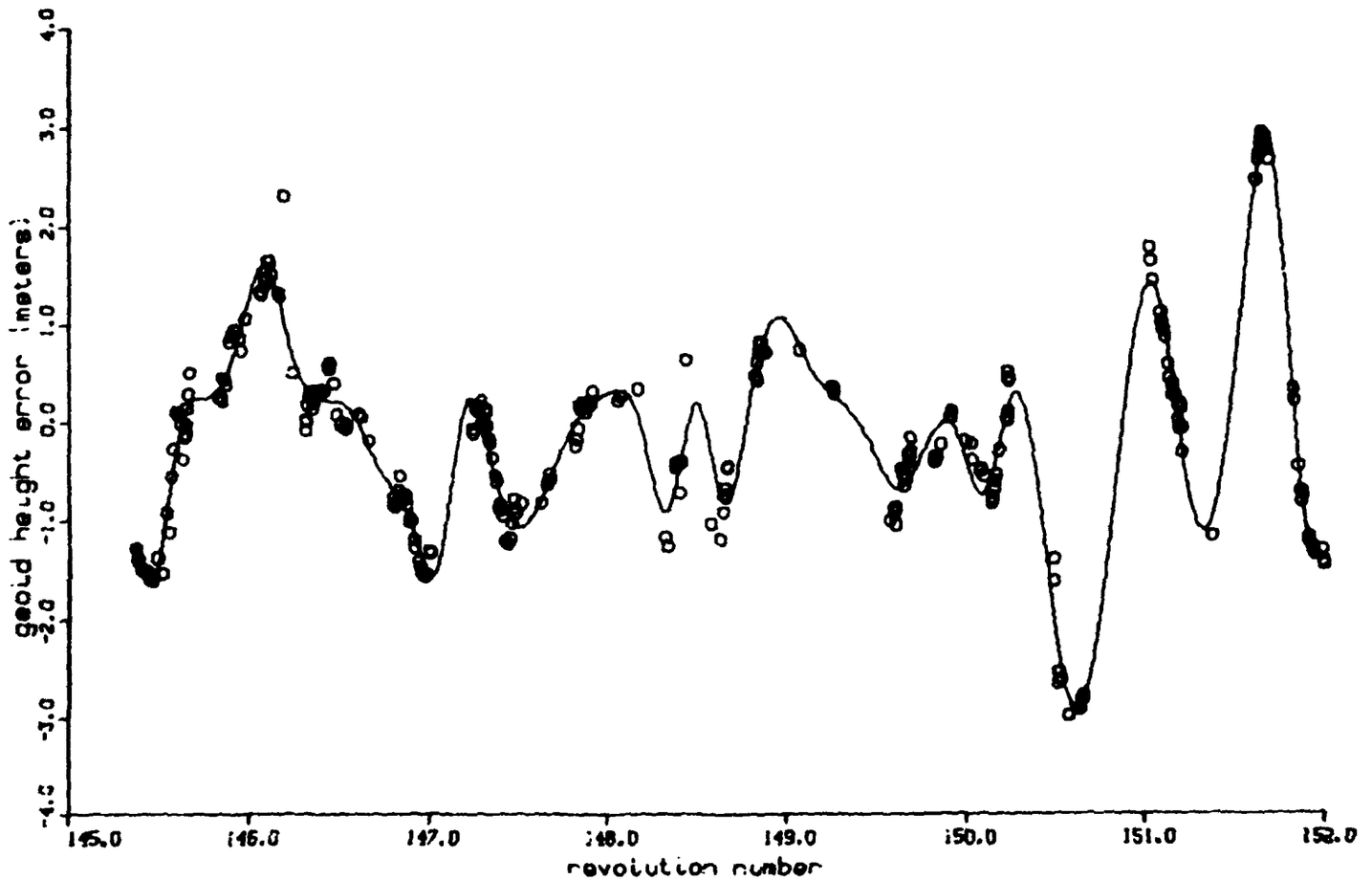


Figure 20. 7 Revolutions of SEASAT Data

Typical Spline Correction (RMS = .19)

(Pseudo-Inverse Solution)

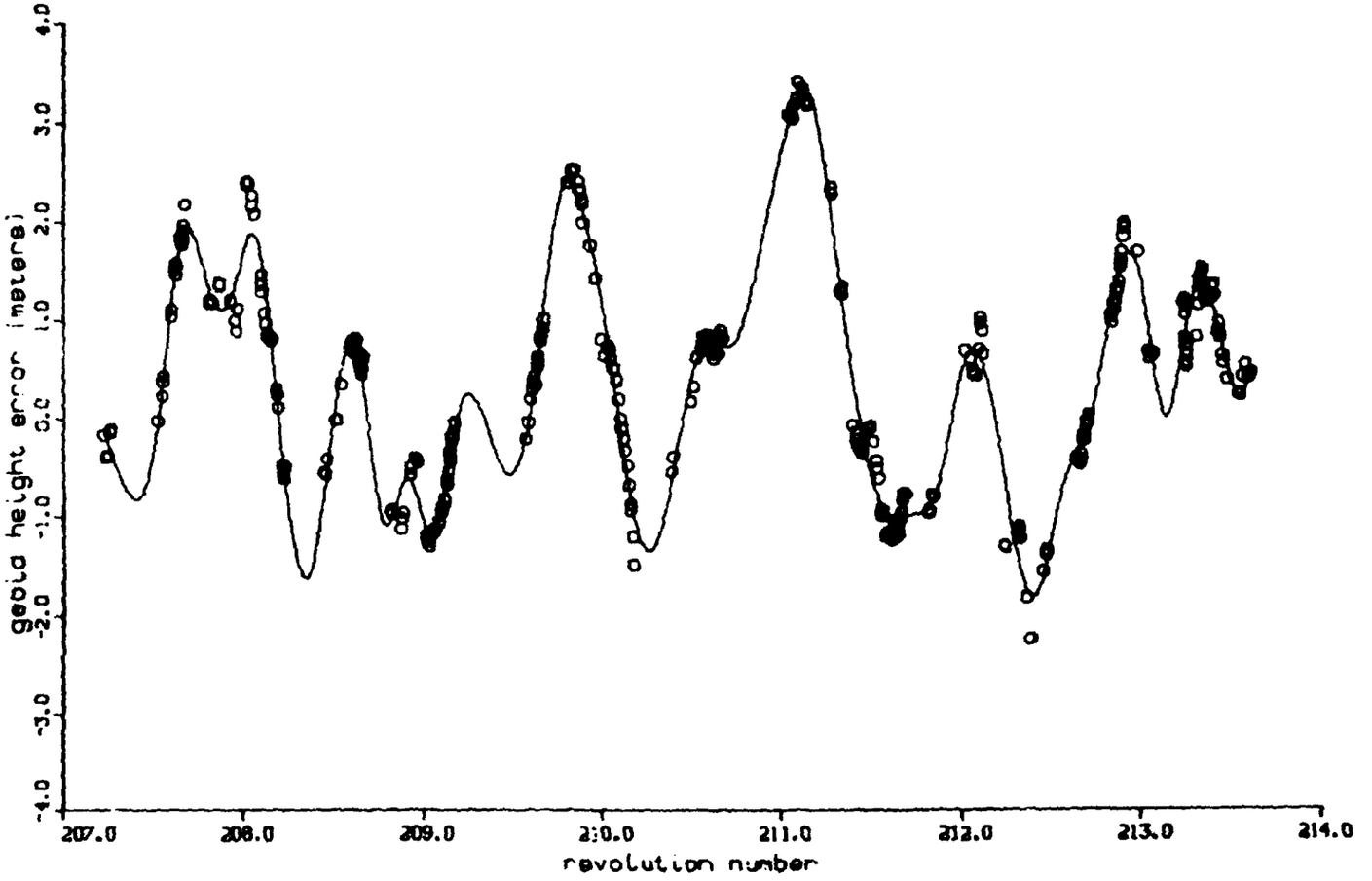


Figure 21. 7 Revolutions of SEASAT Data

Typical Spline Correction (RMS = .17)

(Pseudo-Inverse Solution)

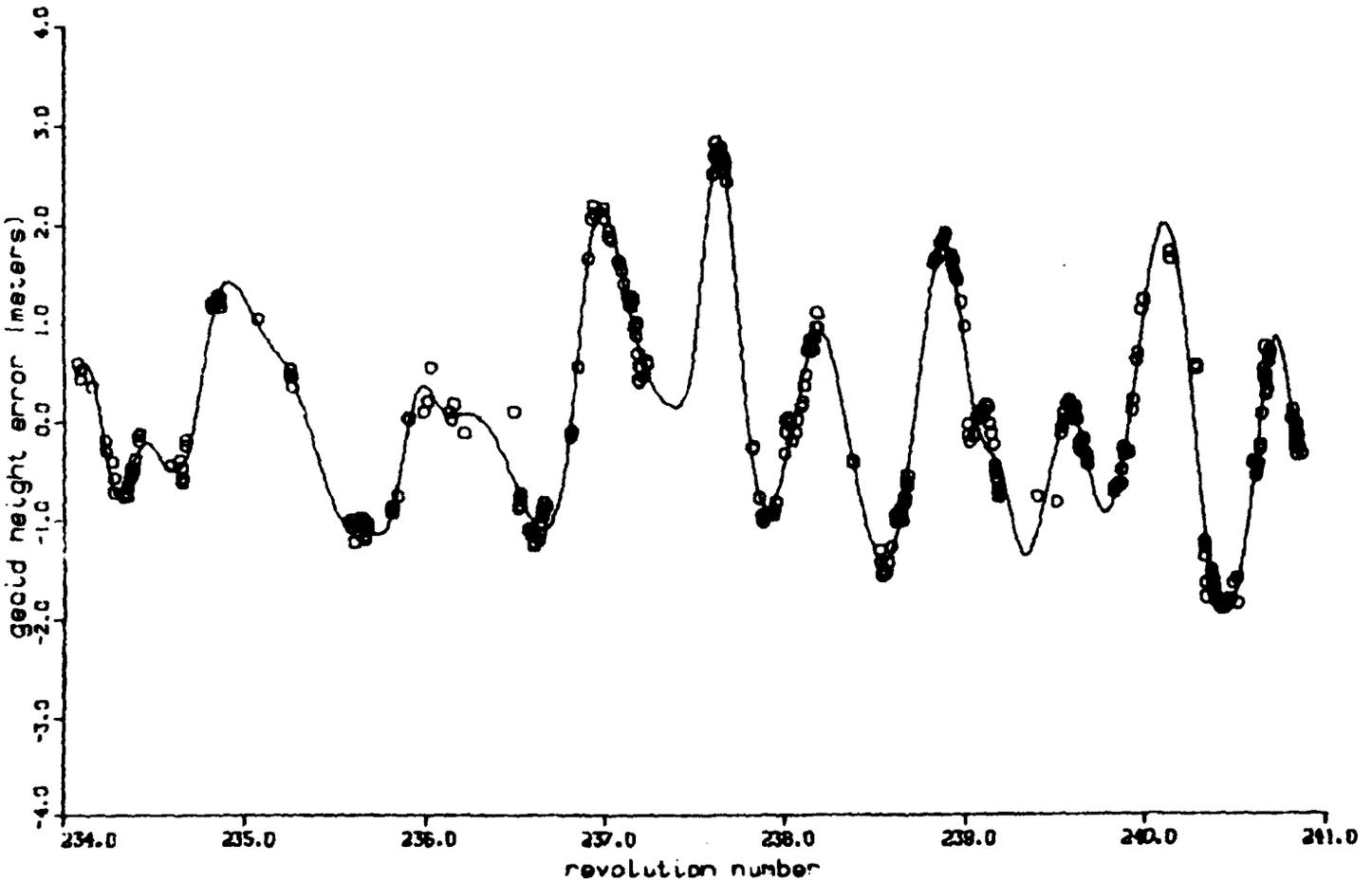


Figure 22. 7 Revolutions of SEASAT Data

Typical Spline Correction (RMS - .19)

(Pseudo-Inverse Solution)

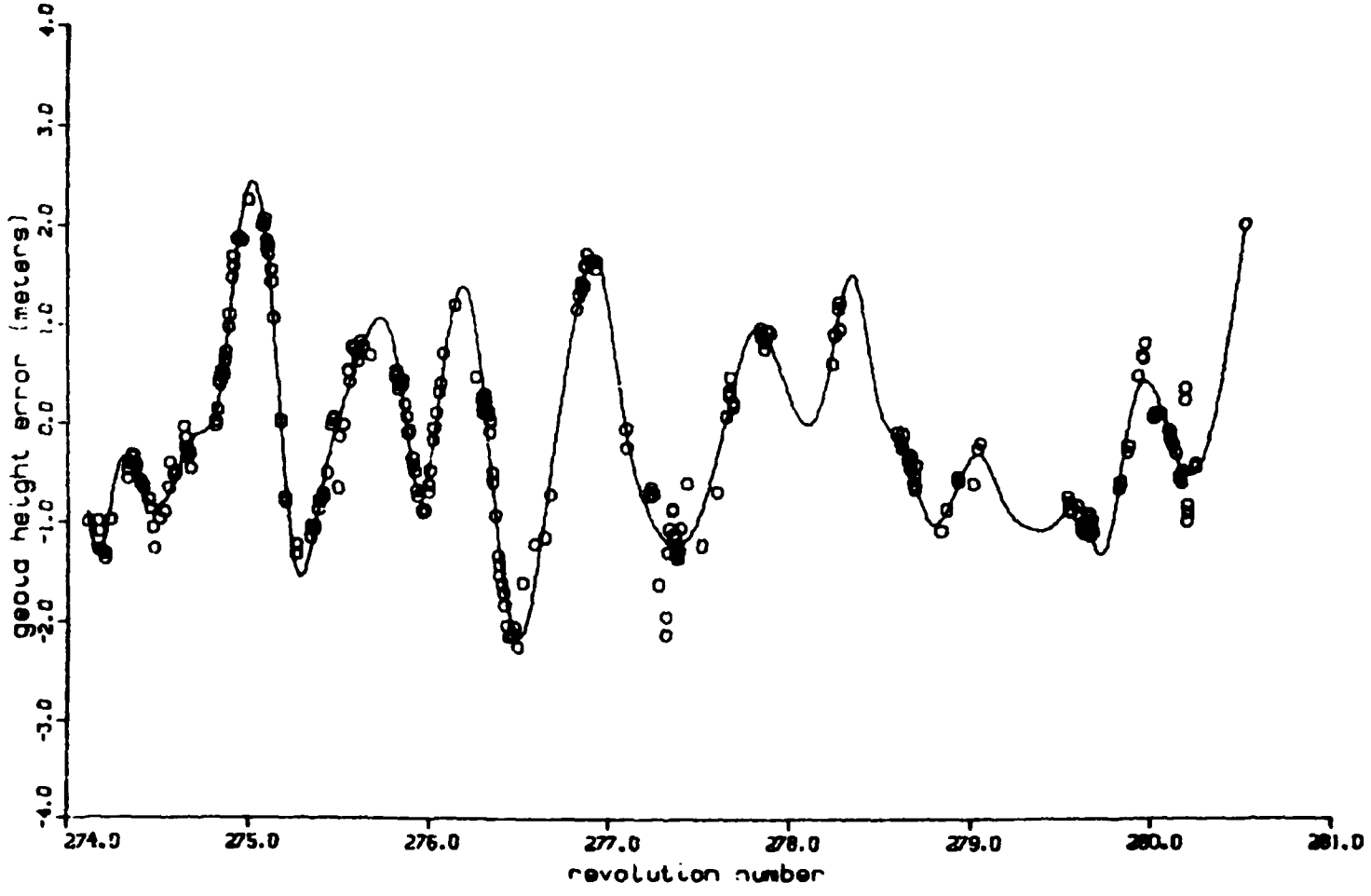


Figure 23. Accuracy Evaluation

RMS1 = .23 RMS2 = .18

—— Typical Spline Correction

---- Fictitious Geoid Height Error

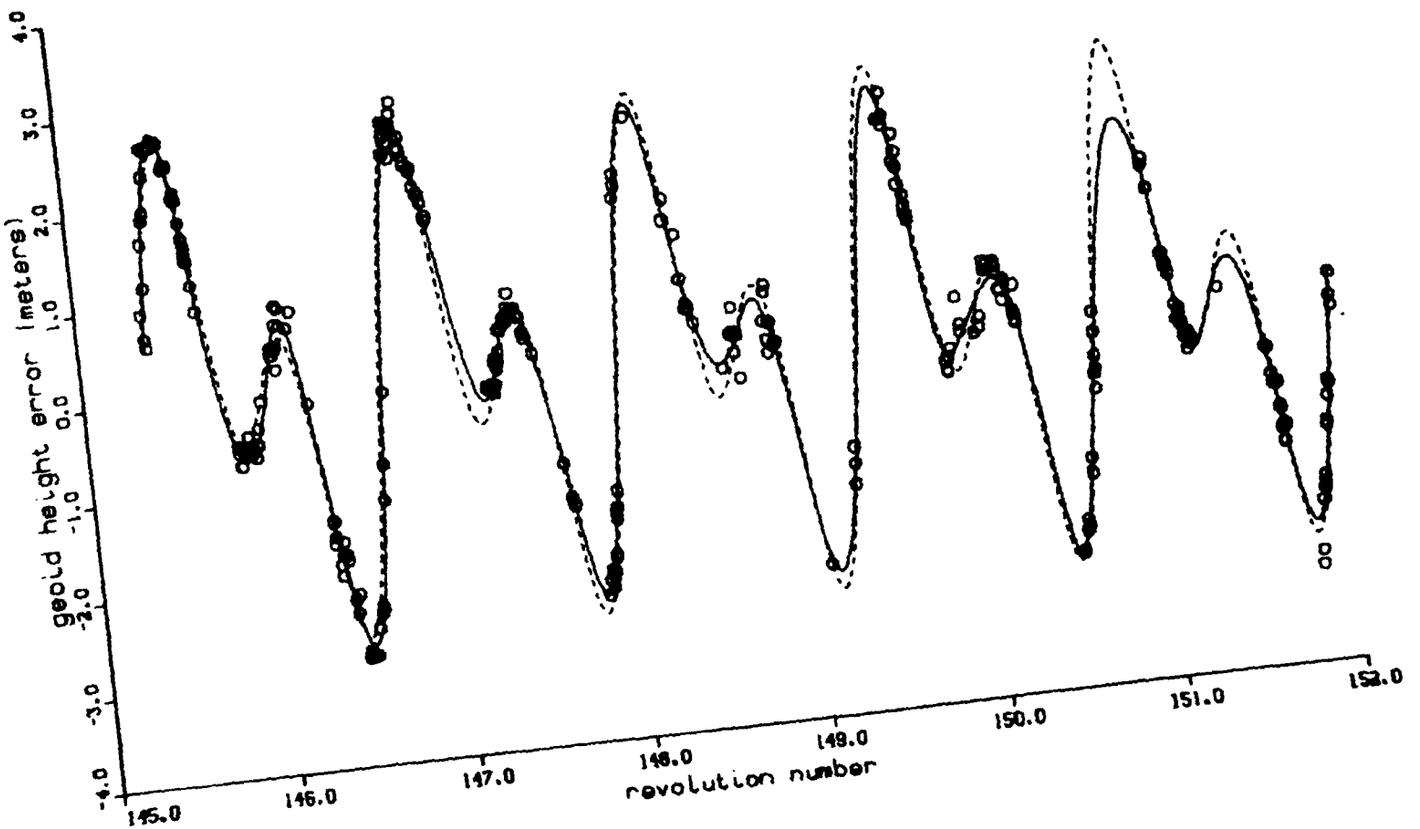


Figure 24. Accuracy Evaluation

RMS1 = .19 RMS2 = .10

—— Typical Spline Correction

---- Fictitious Geoid Height Error

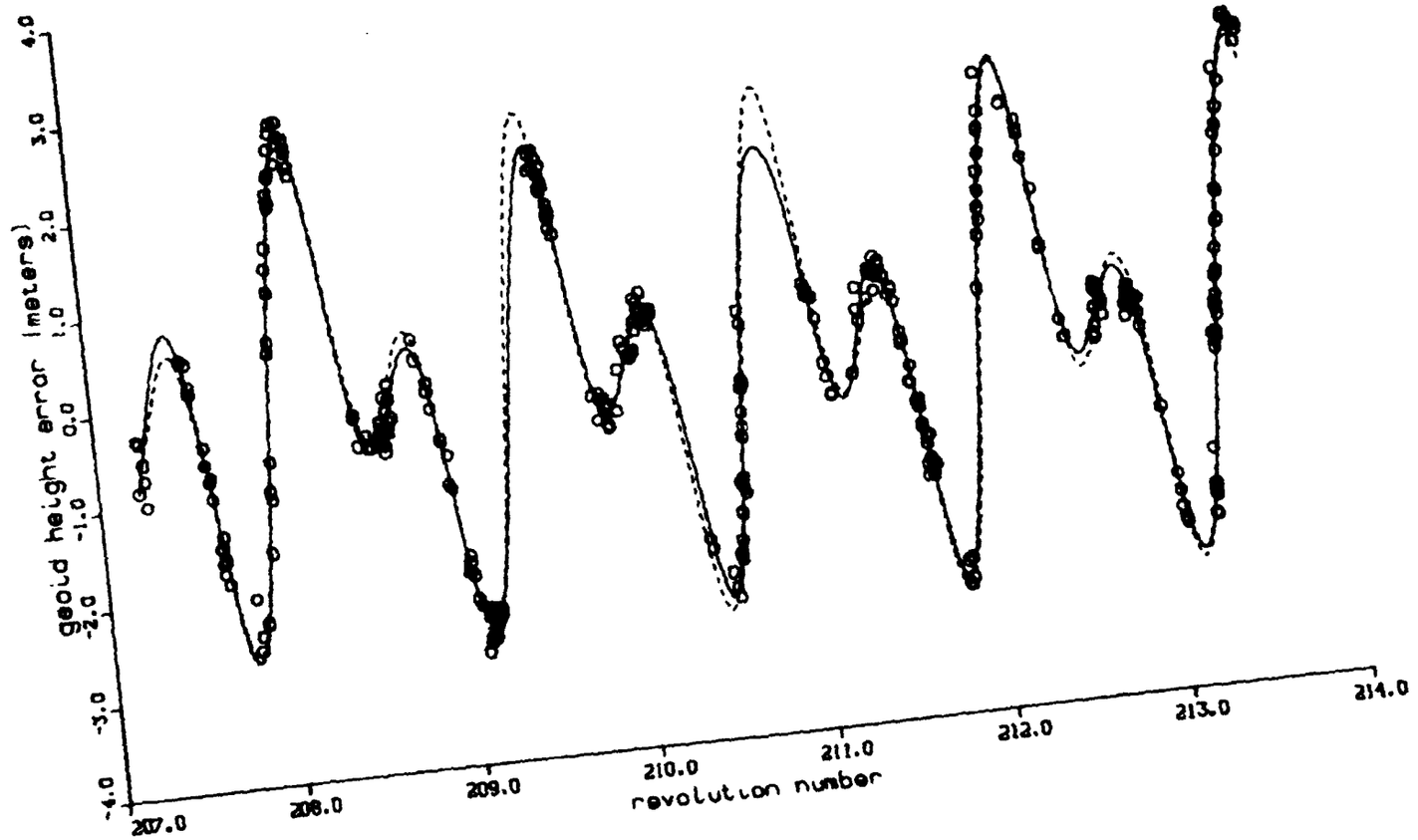


Figure 25. Accuracy Evaluation

RMS1 = .27 RMS2 = .21

—— Typical Spline Correction

---- Fictitious Geoid Height Error

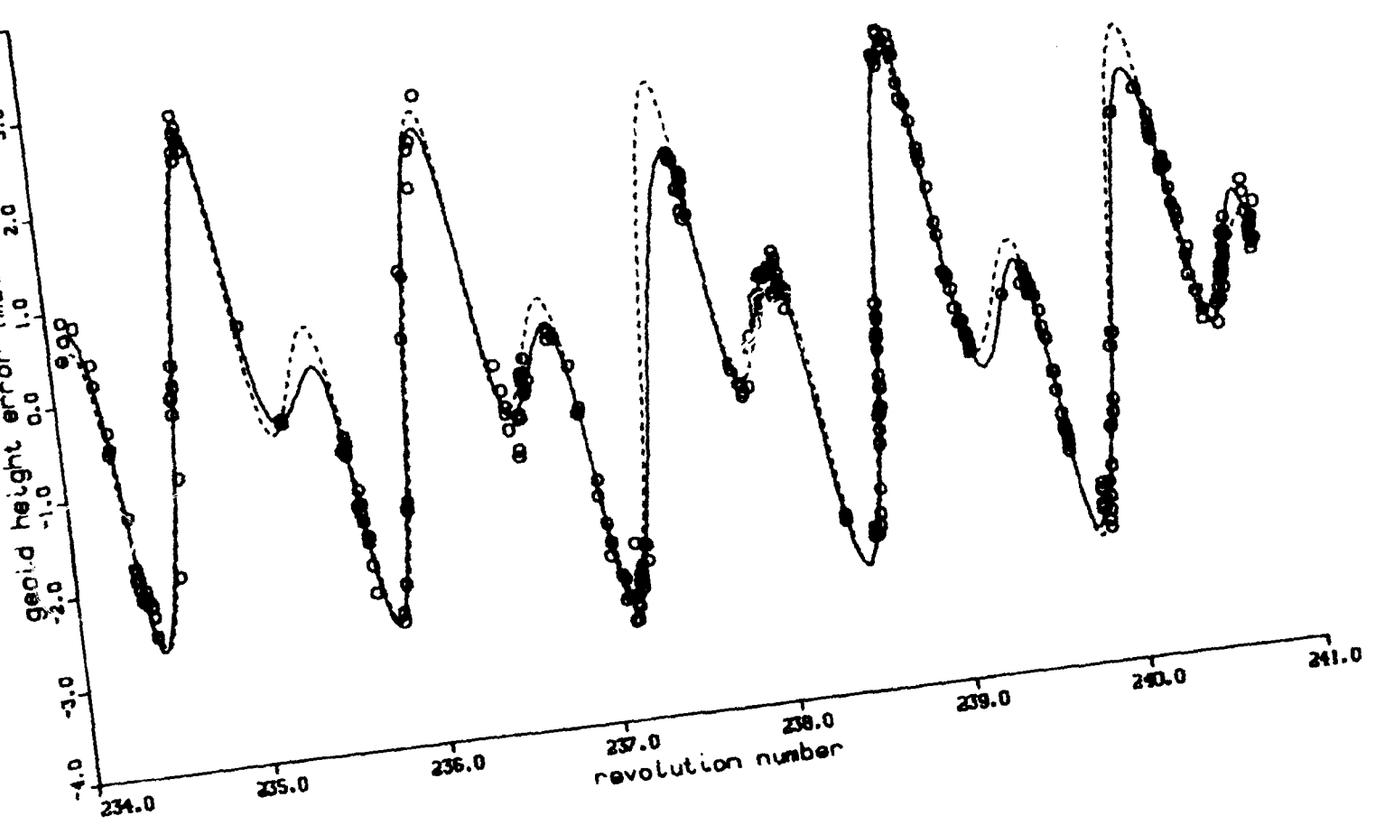
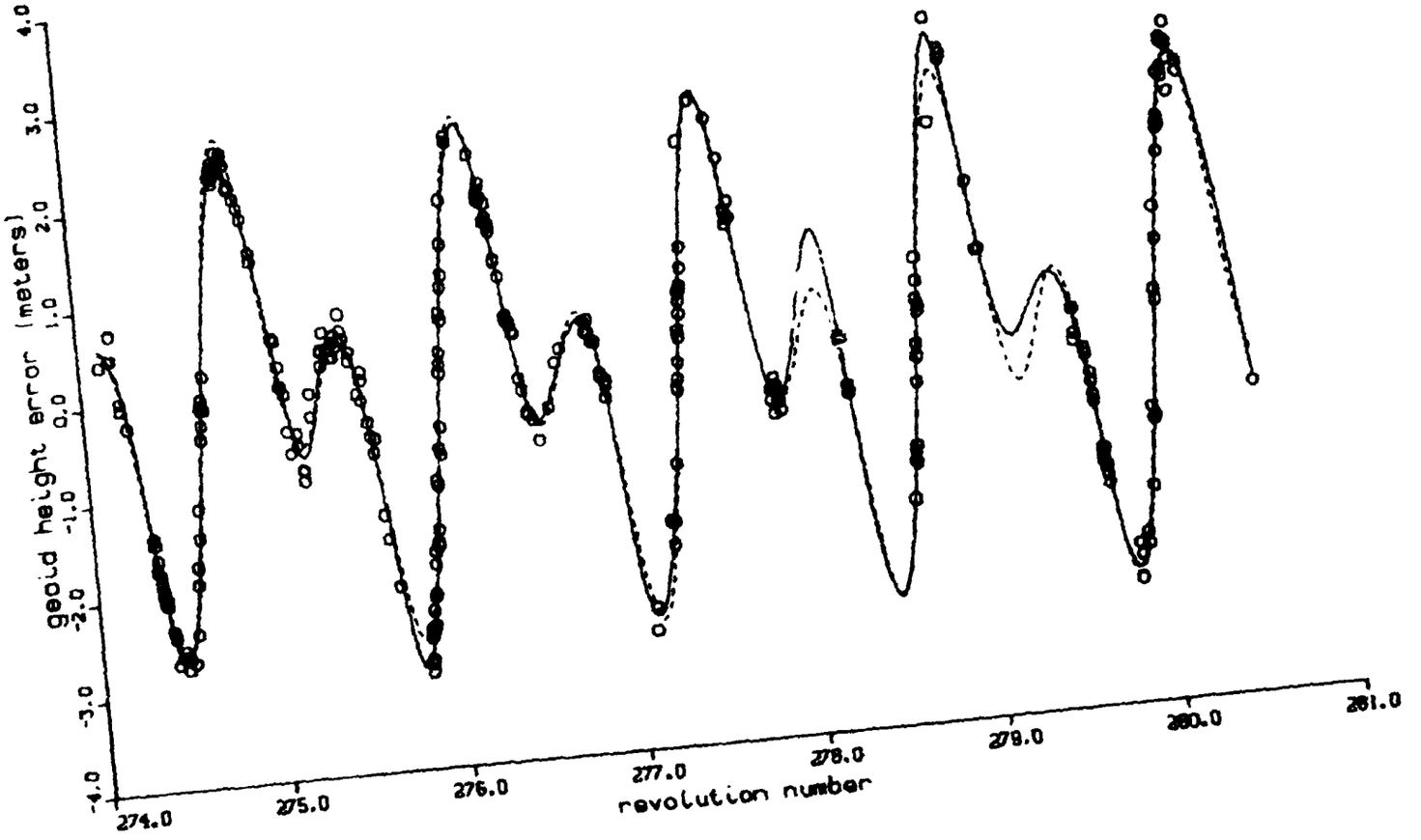


Figure 26. Accuracy Evaluation

RMS1 - .17 RMS2 - .06

—— Typical Spline Correction

---- Fictitious Geoid Height Error



SUMMARY

A new technique has been developed for correcting satellite ephemeris errors indirectly observed from radar altimetry. This new technique differs markedly from previous methods in that a different problem is posed and a different type of solution (discrete as opposed to parameterized-continuous) is obtained. Specifically, a conjugate gradient-projection algorithm is used to find the unbiased, discrete function of minimum weighted variation which produces the geoid height differences observed at the satellite's ground-track intersections.

It is felt that this new procedure represents a substantial improvement over previous efforts. First, it accurately models reality by retaining the concept of one, time-continuous, satellite ground track which repeatedly intersects itself. Secondly, it takes into account correlations between time-contiguous ascending and descending tracks. Thirdly, it allows for an *a posteriori* selection of a functional form for curve fitting purposes. Finally, it requires a minimum amount of computer storage due to the sparseness of the coefficient matrix of the system to be solved.

It has been demonstrated that the method is accurate. Based on the analysis conducted herein, it appears that an altitude correction to an accuracy of 20 centimeters rms is possible.

The real beauty of the technique, however, lies in the fact that it is a general method that can be used to reduce any time-dependent, low-frequency error present in network-type surveys (e.g., oceanographic shipboard and airborne surveys). In the near future, the method will be employed to remove the diurnal variation present in magnetic surveys as well as the nonlinear gravimeter drift present in gravity surveys.

REFERENCES

1. Brace, Kenneth L., Preliminary Ocean Area Geoid from GEOS-3 Satellite Radar Altimetry, Presented at NASA GEOS-3 Investigation, Final Meeting, New Orleans, LA, 18-19 Nov. 1977.

2. Uginčius, Peter, An Iterative Least-Squares Solution for Biases, Bias Rates and Higher-order Terms of Intersecting Altimetry Tracks, NAVSWC/DL Technical Report, 1977.

3. Miele, A., et al., Sequential Gradient-Restoration Algorithm for the Minimization of Constrained Functions - Ordinary and Conjugate Gradient Versions, Journal of Optimization Theory and Applications, Vol. 4, No. 4, 1969.

4. Luenberger, David, Introduction to Linear and Nonlinear Programming, Addison-Wesley Publishing Company, Menlo Park, California, 1973.

5. Cloutier, James R., Generalized Multivariate Splines for Randomly-Spaced Data, NAVOCEANO Technical Report (to appear)

6. Born, George H., Guest Editor, Special Issue on SEASAT Ephemeris Analysis, The Journal of the Astronautical Sciences, Volume XXVIII, No. 4, Oct.-Dec., 1980.

APPENDIX A

Properties of the A Matrix

The A matrix defined in Equations (13) - (16) is symmetric and sparse. The sparseness is due to the relationship between the sets $\{ p_j : j = 1, \ldots, N \}$ and $\{ l_j : j = 1, \ldots, N \}$. This relationship is such that, in any given row (or column) of the A matrix, there exist at most 5 nonzero elements.

The matrix A is also of rank N-1. This is due to the nature of and the relationships between these nonzero elements. From Equation (14) it can be seen that each diagonal element is equal to a sum of weights and is, therefore, positive. Furthermore, from Equations (15) - (16) it can be seen that each nonzero off-diagonal element is equal to the negative of a sum of weights and is, therefore, negative. Not so apparent is the fact that the sum of the absolute values of the off-diagonal elements in any row (or column) is equal to the diagonal element in that row (or column), i.e.,

$$a_{m,m} = \sum_{\substack{n=1 \\ n \neq m}}^{N} \left| a_{m,n} \right| = \sum_{\substack{n=1 \\ n \neq m}}^{N} \left| a_{n,m} \right| \qquad m = 1, \ldots, N \tag{A-1}$$

Theorem 1. The matrix A is positive semidefinite.

Proof: From Gershgorin's Theorem, every eigenvalue λ of A must satisfy at least one of the inequalities

$$\left| \lambda - a_{m,m} \right| \leq \sum_{\substack{n=1 \\ n \neq m}}^{N} \left| a_{m,n} \right| \qquad m = 1, \ldots, N \tag{A-2}$$

Substituting (A-1) into (A-2) yields

$$\left| \lambda - a_{m,m} \right| \leq a_{m,m} \qquad m = 1, \ldots, N \tag{A-3}$$

which implies that

$$0 \leq \lambda \leq 2a_{m,m} \tag{A-4}$$

Hence, A is positive semidefinite.

Theorem 2. The matrix A is positive definite on the subspace $M_1 = \{ \vec{x} : \vec{x}^T \vec{d} = 0 \}$

Proof: A is positive semidefinite (and thus can be split into $\tilde{A}^T \tilde{A}$ where $\tilde{A}$ is the square root matrix). Hence, we only need to show that for nonzero $\vec{x} \in M_1$, $A\vec{x} \neq \vec{o}$.

Suppose that $\vec{x}^T \vec{d} = 0$ and $A\vec{x} = \vec{0}$. Then $\vec{x} \in N(A)$, where N(A) is the null space of A. Now $\vec{d} \in N(A)$ since $A\vec{d} = \vec{0}$ and N(A) is of dimension 1 since A is of rank N-1. Thus $\vec{x} = a\vec{d}$ for some $a \neq 0$. This implies that $\vec{x}^T \vec{d} = a\vec{d}^T \vec{d} = aN \neq 0$ which contradicts the fact that $\vec{x} \in M_1$. Hence, A is positive definite on M_1.

Theorem 3. The matrix A is positive definite on the subspace $M_2 = \left\{\vec{x}: C^T \vec{x} = \vec{0}\right\}$

Proof: $M_2 \subset M_1$

DISTRIBUTION LIST

CNO	7
ASN(R,E&S)	2
ONR	5
NRL	3
NORDA	5
NAVOBS	2
COMNAVOCEANCOM	2
USNA	1
NAVPGSCOL	1
NAVSWC	11
DSSPO	7
DMA	3
DMAAC/STT	3
DMAH/TC	3
DMAODS	1
NOAA/NOS	1
NOAA/NOS/NGS	4
NOAA/PMEL	1
NASA/GSFC	1
NASA/WFC	1
ATL	1
AFGL/Hanscom	1
DTIC	12
CU/LDGO	1
CIT/JPL	5
OSU/DGS	1
ODU/DGS	1
UTA/DAEEM	3
JHU/APL	3
MIT/DEPS	1
UC/SIO	1
UM/RSMAS	1
UH/DO	1
AEROSPACE	1
BATTELLE	2
FEAI	1
DRAPER	4
LOCKHEED	1
SINGER-KEARFOTT	1
GE/OS	1
WHOI	1
SAO	1
ASC	1